AFRL-IF-RS-TR-2002-196
Final Technical Report
August 2002

THE USE OF END-TO-END MULTICAST MEASUREMENTS FOR CHARACTERIZING INTERNAL NETWORK BEHAVIOR

University of Massachusetts

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J172

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

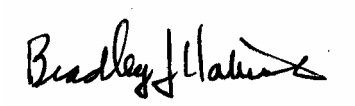
The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-196 has been reviewed and is approved for publication

APPROVED:



BRADLEY J. HARNISH
Project Engineer

FOR THE DIRECTOR:

WARREN H. DEBANY, Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Aug 02	Final Jul 98 – Jan 02

4. TITLE AND SUBTITLE
THE USE OF END-TO-END MULTICAST MEASUREMENTS FOR CHARACTERIZING INTERNAL NETWORK BEHAVIOR

5. FUNDING NUMBERS
C - F30602-98-2-0238
PE - 62110E
PR - G202
TA - 00
WU - 01

6. AUTHOR(S)

Don Towsley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Massachusetts
Department of Computer Science
140 Governors Drive
Amherst, MA 01003

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFGA
525 Brooks Rd
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2002-196

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Bradley J. Harnish, IFGA, 315-330-1884, harnishb@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

This is the Final Technical Report for MINC: Multicast-based Inference of Network-internal Characteristics. The primary aim of the MINC project was to develop a statistical foundation on which to develop and evaluate end-to-end based tools for inferring internal network behavior. The project accomplished two goals. First, a series of techniques based on maximum likelihood estimation was developed to infer link-level loss and delay behavior, and internal topological structure based on end-to-end multicast and unicast observations. Briefly, these techniques exploit the correlation inherent in multicast-based measurements to produce accurate estimates. Second, measurement software required to produce the necessary observations and a web-based analysis and visualization tool were developed and made available to the community. The measurement software can be included in any multicast application as well as used on the National Internet Measurement Infrastructure. The analysis/visualization tool allows the user to obtain and visualize time varying loss rate estimates.

14. SUBJECT TERMS
Multicast, Network topography, End-to-end measurement, Maximum likelihood estimation, Networking modeling

15. NUMBER OF PAGES
241

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures

1 Introduction

The primary goal of the MINC project was to complete the fundamental research required to develop an inference methodology that would, for the first time, allow end-systems and other network elements to determine the performance characteristics (such as loss, delay, and available bandwidth) of both individual links and end-end paths within an internetwork. The goal was a methodology that did not require the active participation of the links and routers being characterized. The MINC project also had as a goal to develop measurement tools that would run on the NIMI (the National Internet Measurement Infrastructure), and to devlelop analysis tools to apply to the measurements.

The main outcomes of the project were the following

- **Multicast-based network inference techniques:** We developed statistically rigorous estimation techniques to determine link loss and delay, and to identify the multicast tree topology. The techniques use the set of receiver traces of end-end path performance in a multicast tree and exploit the correlation that multicast traffic inherently contains to derive link-level performance metrics.
- **Unicast-based network inference techniques:** We also developed similar techniques, which rely on unicast measurements.
- **Measurement layout techniques:** We developed efficient algorithms for selecting and laying out low cost multicast distribution trees for the purpose of inferring the behavior of a target set of network links.
- **Measurement, analysis, and visualization tools:** We developed end-to-end measurement tools that run on NIMI and a validated, web-based tool, MINT (Multicast Inference Network Tool) that allows an analyst to analyze and visualize internal network behavior.

We describe each of these outcomes in the remainder of this report. Prior to this, we motivate the need for our project.

2 Motivation

As the Internet grows in size and diversity, its internal performance becomes ever more difficult to measure. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data. End-to-end measurements using unicast traffic do not rely on administrative access privileges, but it is difficult to infer link-level performance from them and they require large amounts of traffic to cover multiple paths. There is, consequently, a need for practical and efficient procedures that can take an internal snapshot of a significant portion of the network.

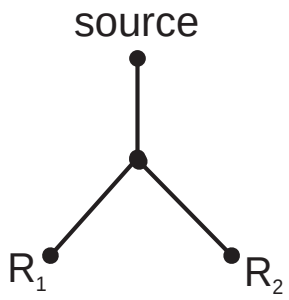


Figure 1: A tree connecting a sender to two receivers.

The focus of the MINC project has been the development of measurement techniques that address these problems. MINC (*Multicast Inference of Network Characteristics*) uses end-to-end multicast measurements to infer link-level loss rates and delay statistics by exploiting the inherent correlation in performance observed by multicast receivers. These measurements do not rely on administrative access to internal nodes since they are done between end hosts. In addition, they scale to large networks because of the bandwidth efficiency of multicast traffic.

Focusing on loss for the moment, the intuition behind packet loss inference is that the event of the arrival of a packet to a given internal node in the tree can be inferred from the packet's arrival at one or more receivers descended from that node. Conditioning on this latter event, we can determine the probability of successful transmission to and beyond the given node. Consider, for example (Figure 1) a simple multicast tree with a root node (the source), two leaf nodes (receivers R_1 and R_2), a link from the source to a branch point (the shared link), and a link from the branch point to each of the receivers (the left and right links). The source sends a stream of sequenced multicast packets through the tree to the two receivers. If a packet reaches either receiver, we can infer that the packet reached the branch point. Thus the ratio of the number of packets that reach both receivers to the total number that reached only the right receiver gives an estimate of the probability of successful transmission on the left link. The probability of successful transmission on the other links can be found by similar reasoning.

The remainer of this report is organized as follows. We describe the MINC multicast methodology in Section 3, its adaptation to the use of unicast in Section 4, and the tree layout methodology (Section 5). Following this, we describe the measurement methodology and the analysis tool, MINT (Section 7). Section 8 offers some conclusions.

3 Statistical Methodology

MINC works on *logical* multicast trees, i.e. those whose nodes are identified as branch points of the physical multicast tree. A single logical link between nodes of the logical multicast tree may comprise more than one physical link. MINC infers composite properties of the logical links. Henceforth when we speak of trees we will be speaking of logical multicast trees.

3.1 Loss Inference

We model packet loss as independent across different links of the tree, and independent between different probes. Thus, the loss model associates with each link k in the tree, the probability α_k that a packet reaches the terminating node of the link, also denoted by k, given that it reaches the parent node of k. The link loss probability is, then, $(1 - \alpha_k)$. Each receiver records the outcome of each probe sent by the source, i.e., whether it is received or not. The α_k can be expressed directly as a function of the probabilities of all possible outcomes of success and loss of a probe at each receiver. An experiment consists of a series of probes transmitted from the source. The outcome of each probe at each receiver is recorded and the link probabilities are inferred by the estimators $\widehat{\alpha}_k$ obtained by using the actual frequencies of the outcomes. [4] contains a detailed description and analysis of the inference algorithm.

The estimators $\widehat{\alpha}_k$ exhibit several desirable statistical properties. It was shown in [4] that $\widehat{\alpha}_k$ is the Maximum Likelihood Estimator (MLE) of α_k when sufficiently many probes are used. The MLE is defined as the set of link probabilities that maximizes the probability of obtaining the observed outcome frequencies. The MLE property in turn implies two further properties of $\widehat{\alpha}$, namely

1. *consistency*: $\widehat{\alpha}_k$ converges to the true value α_k almost surely as the number of probes n grows to infinity, and

2. *asymptotic normality*: the distribution of the quantity $\sqrt{n}(\widehat{\alpha}_k - \alpha_k)$ converges to a normal distribution as n grows to infinity.

The latter property implies that the probability of an error of a given size in estimating a link probability goes to zero exponentially fast in the number of probes.

The computation of the $\widehat{\alpha}_k$ is performed recursively on the tree; the computational cost is linear in the number of probes and number of nodes in the tree. Details of this algorithm are found in [4], which can be found in the Appendix.

Often we are faced with the problem of inferring the link loss behavior for a collection of multicast trees. We have developed two approaches for handling this problem. The first is to treat each tree separately and obtain the MLE for each segment cotained within each tree. If a segment is found in two or more trees, then we return the *minimum variance weighted average (MVWA)*, an estimate equal to the weighted average of the MLEs obtained from the different trees. Here the weights are taken to be proportional to the variances in the link estimates.

The second approach consists of applying the *expectation maximization (EM)* algorithm [16] to the entire collection of trees. Briefly, the EM algorithm calculates the maximum lieklihood estimate of the link probabilities in an iterative fashion. Our experience with both the MVWA and EM algorithms suggests that much of the time there is little difference in the quality of the estimates that they return. However, when there are either few observations or the numbers of observations vary widely from tree to tree, the EM algorithm

produces a more accurate estimate. Details of these two approaches can be found in [3], which can be found in the Appendix.

Finally, situations arise where observations may be missing from different subsets of receivers in a tree. Such a situation is easily handled using the EM algorithm to fill in the missing observations with estimates based on the observations that are available. We have observed from simulation studies that the quality of the estimates produced when observations are missing is close to that for the case of complete observations unless the percantage of missing observations exceeds 70% to 80%. Details of the algorithm and its evaluation can be found in [12], also found in the Appendix.

3.2 Delay Distribution Inference

A generalization of the loss inference methodology allows one to infer per link delay distributions. More precisely, we infer the distribution of the variable portion of the packet delay, what remains once the link propagation delay and packet transmission time are removed. Packet link delays are modeled as discrete random variables that can take one of a finite number of values, independent between different packets and links. The model is specified by a finite set of probabilities $\alpha_k(t)$ that a packet experiences delay t while traversing the link terminating at node k, with infinite delay interpreted as loss.

When a probe is transmitted from the source, we record either the time taken by a probe to reach each receiver or that the probe was lost. As with the loss inference, a probabilistic analysis enables us to relate the $\alpha_k(t)$ to the probabilities of the outcomes at the receivers. We infer the link delay probabilities by the estimators $\widehat{\alpha}_k(t)$ obtained by using instead the actual frequencies of the outcomes arising from the dispatch a number of probes. In [15], it was shown that the corresponding estimator$\widehat{\alpha}(\cdot)$ of the link delay distributions is strongly consistent and asymptotically normal. [15] is found in the Appendix.

3.3 Delay Variance Inference

The delay variance can be directly estimated Consider the binary topology of Figure 1. Let D_0 be the packet delay on the link emanating from the source, and $D_i,\ i = 1, 2$, the delay on the link terminating at receiver i. The end-to-end delays from the source to leaf node $i = 1, 2$, is expressed as $X_i = D_0 + D_i$. A short calculation shows that, with the assumption that the D_i are independent, $\mathrm{Var}(D_0) = \mathrm{Cov}(X_1, X_2)$. Thus the variance of the delay D_0 can be estimated from the measured end-to-end delays from the source to the leaves. This approach has been generalized to estimate link delay variances in arbitrary trees. Details can be found in [13], which is is contained within the Appendix.

3.4 Topology Inference

In the loss inference methodology described above, the logical multicast tree was assumed to be known in advance. However, extensions of the method enable inference of an unknown multicast topology from

end-to-end measurements. We describe briefly three approaches.

The first, which relies on loss-based grouping was suggested in [21], in the context of grouping multicast receivers that share the same set of network bottlenecks from the source. The loss estimator of Section 3.1 estimates the shared loss to a pair of receivers, i.e., the composite loss rate on the common portion of the paths from the source, irrespective of the underlying topology. Since this loss rate is larger the longer the common path in question, the actual shared loss rate is maximized when the two receivers are siblings.

A binary tree can be reconstructed iteratively using this approach. Starting with the set of receiver nodes R, select the pair of nodes j, k in R that maximizes the estimated shared loss, group them together as the composite node. Iterate on this and the set remaining nodes from R until all are grouped. The algorithm is consistent: the probability of correct identification converges to one as the number of probes grows; see [11]. General (i.e. non-binary) trees can be inferred by using this algorithm and then transforming the resulting binary tree by pruning links with inferred loss rates less than some threshold ε. We refer to this as the *binary loss tree with pruning (BLTP)* algorithm.

A second, Bayesian approach assumes that the topology comes from a known prior distribution. It then identifies the topology that minimizes a posterior risk function associated with topologies. This approach requires the least number of observations in order to classify the topology, provided that the topology comes from the known prior distribution.

The third, a direct ML approach, calculates the maximum likelihood of the measured outcomes over all possible α_k. The topology that maximizes this quantity is chosen to be our estimate. This classifier is consistent [11].

Our experience with these three approaches is that the Bayesian algorithm is slightly more accurate than the BLTP algorithm, but at the cost of significant computational complexity. The direct ML approach does not do as well as either of the other algorithms. Details of these algorithms and their evaluation can be found in [11]. This publication can be found in the Appendix.

The loss-based grouping algorithm approach can be extended by replacing shared loss with any function on the nodes that (i) increases on moving further from the source; and (ii) whose value at a given node can be consistently estimated from measurements at receivers descended from that node. The mean and variance of the cumulative delay from the source to a given node exhibit these properties. Hence, multicast end-to-end delay measurements can also be used to infer the multicast topology. Details of this approach can be found in [9], which is contained in the Appendix.

Last, We have combined the use of losses and delays to develop a hybrid grouping algorithm that combines the best of BLTP and a delay based algorithm. The basic idea is as follows. At each step where two nodes are to be combined, candidates are selected based on losses and on delays. The probabilities of each of these being in error are estimated and the candidates that yield the lowest error are chosen. In other words, the selection is determined by losses if this leads to a lower probability of error and by delays otherwise.

This approach is described in [8], which is contained in the Appendix.

4 Unicast-based Methods

At the start of this project we had every expectation that multicast woud become pervasive throughout the Internet. Unfortunately, its deployment has proceeded very slowly. Consequently, we turned our attention to adapting our techniques for use with end-to-end unicast measurements.

In this section, we describe our work to adapt the multicast inference techniques described in Section 3 to perform inference of internal network characteristics from unicast end-to-end measurements. The data for the inference comprises measured end-to-end loss of unicast probes sent from a source to a number of destinations. This is used to infer the loss and delay characteristics of each logical link of the source tree joining the source to the destinations, i.e., of the composite paths between its branch points.

The idea is to construct composite probes of unicast packets whose collective statistical properties closely resemble those of a multicast packet. We shall speak of **striping** a group of unicast packets across a set of destinations. This entails dispatching the packets back-to-back from a source, each packet potentially having a different destination address. Our premise is that when the duration of network congestion events exceeds the temporal width of the stripe, packets should have very similar experience of the network upon traversing common portions of the paths to their destinations. If the experiences were identical, the packets from a stripe that attempt to traverse a given link would either all be lost, or encounter identical delay. Hence the packet loss and delays on a given link would be perfectly correlated within a stripe; the composite probe would have the same statistical properties as a notional multicast packet that followed the same source tree. In this case the methods presented in Section 3 could be applied immediately to infer the per link loss and delay statistics of the logical links source tree.

However, correlations within stripes may be less than perfect in practice. This is because congestion events may not affect packets uniformly, subjecting stripes to dispersion as they travel through a network. Some mechanisms by which this can happen are the following. Packet loss will not be uniform during loss events that are narrower than the stripe, or those that start or stop while the stripe is in progress. Furthermore, delays will vary due to interleaving of background traffic, e.g., when moving from a low to a high capacity link. Although such effects should be small for sufficiently narrow stripes, they will be cumulative. Packet-dropping on the basis Random Early Detection (RED) [17] is another mechanism by which packet loss may become decorrelated. It remains to be seen whether this mechanism will be widely deployed in communications networks. On the other hand, the use of RED to merely mark packets will not break correlations.

This motivated the following four investigations:

(i) determining the magnitude of imperfect correlations through experiments on real networks;

(ii) calculating their likely impact on the accuracy of inference via methods that assume perfect correlations;

(iii) adopting measurement procedures that reduce the impact of imperfect correlations;

(iv) verifying the accuracy of the approach in simulations.

We extended the packet loss model of Section 3 by incorporating an additional parameter for each link that describes the correlation of loss between different packets of the same stripe. This is done for binary stripes, i.e., those comprising two packets with different destination addresses. These additional parameters cannot themselves be determined by end-to-end measurements, at least not without additional assumptions relating them to each other, or to the existing loss rate parameters.

By constructing appropriate stripes of composite probes and selecting subsets of these probes for inference, we are able to enhance correlations within data used for inference. This is possible when packet transmissions are correlated in the sense that a given packet in a stripe is more likely to be transmitted when other of its packets are known to have been transmitted. By conditioning on the measurable event that nearby packets have been transmitted, we raise the likelihood of transmission of a given packet closer to 1. By sending the striped packets to diverse addresses, we can infer the properties of internal network paths from the measurements.

We evaluated the proposed methods through through measurements over the Internet and through simulation. We collected end-to-end measurements on the National Internet Measurement Infrastructure (NIMI) [19] from a diverse set of Internet paths. We transmitted stripes between pairs of end-hosts and verified that their packet loss statistics were consistent with the correlation assumptions that underlie the method. (These stripes were different from those defined above, since all packets in the stripes were sent to the same destination) We also estimated the likely accuracy that would be obtained by stripe-based inference in the network.

We supported the measurement work through network level simulation with `ns` [20]. By instrumenting the simulation we traced the origin of end-to-end behavior to network internal characteristics. This allowed us first to exhibit the correlation properties of packet within stripes as they are transmitted across individual links in the network (rather than just the end-to-end properties), and second to compare the inferred link loss rates with actual link loss rates. For the most accurate choice of striping method we find the typical absolute error in loss rate inference to be below 1%.

Details of this extension to unicast is found in [14], also contained within the Appendix.

5 Tree Layout

In the previous sections, we have described algorithms for inferring internal network behavior. In this section we describe algorithms developed during the project for choosing the trees over which to perform

measurements.

A network is represented by a directed graph $N = (V, E)$ where V and E denote the set of nodes and links within N respectively. Our interest is in multicast trees embedded within N. Let $S \subseteq V$ be a set of possible multicast senders, and $R \subseteq V$ be a set of possible multicast receivers. Let $T = (V_T, E_T)$ denote a directed (multicast) tree with a source s_T and a set of leaves r_T. We require that $s_T \in S$ and $r_T \subseteq R$. Let A be a routing algorithm that produces the distribution tree $A(s, r)$ between a source $s \in S$ and receiver set $r \subseteq R$. Let $\mathcal{T}(A, S, R) = \{A(s, r) : s \in S, r \subseteq R/\{s\}\}$, i.e., $T(S, R)$ is the set of all possible multicast distribution trees within the network N with sources from S and receiveers from R. A.

Associated with a multicast tree $T \in \mathcal{T}(S, R)$ is a cost

$$C(T) = C_T^0 + \sum_{l \in E(T)} C_l, \tag{1}$$

where the first term can be thought of as a "per tree cost" and the second is a "per link cost". For example, IP multicast requires each multicast router to maintain per flow state. This is accounted for by the per tree cost. The per link cost is the cost for sending probe packets through a link. The two problems of interest to us are as follows:

Multicast Tree Identifiability Problem (MTIP). Given a set of multicast trees $\Psi \subseteq \mathcal{T}(S, R)$, and a set of links $L \subseteq E$, is L identifiable by the set of trees Ψ?

Minimum cost Multicast Tree Cover Problem (MMTCP). Given $S, R \subseteq V$, $L \subseteq E$ and L is identifiable by $\mathcal{T}(S, R)$, what is the minimum cost subset of $\mathcal{T}(S, R)$ sufficient to cover L? In other words, find $\Psi \subseteq \mathcal{T}(S, R)$ that covers L and minimizes

$$C(\Psi) = \sum_{T \in \Psi} C(T)$$

We have developed the following during the project.

- We developed efficient algorithms for solving the identifiability problems.
- We established that the cover problem is NP-hard and that in some cases, finding an approximation within a certain factor of optimal is also NP-hard. Thus, we also propose several heuristics and show through simulation that a greedy heuristic that iteratively combines trees containing a small number of receivers performs reasonably well.
- We developed polynomial time algorithms that find optimal solutions for a restricted class of network topologies, including trees. This algorithm can be used to provide a heuristic for sparse, tree like networks. This heuristic is also shown through simulation to perform well.
- We applied our techniques to the vBNS and Abilene network as well as randomly generated networks, showing the effectiveness of the different heuristics.

Details of this work can be found in [2], which can be found in the Appendix.

6 Experimental Results

In this section we briefly describe our efforts to validate the MINC methodology. Section 6.1 contains a description of the results of a measurement study in which we collected end-to-end loss traces from the MBone and validated the results from inferences of loss rates collected using the Internet tool `mtrace`. Section 6.2 contains a description of the results from more detailed simulation studies of both loss and delay.

6.1 Measurement Experiments

To validate MINC under real network conditions, we performed a number of measurement experiments on the MBone, the multicast-capable subset of the Internet. Across our experiments we varied the multicast sources and receivers, the time of day, and the day of the week. We compared inferred loss rates to directly measured loss rates for all links in the resulting multicast trees. The two sets of quantities agreed closely throughout.

During each experiment, a source sent a stream of 40 byte, sequenced packets every 100 miiliseconds to a multicast group consisting of a collection of receivers over the course of one hour. The resulting traffic stream placed less than 4 Kbps of load on any one MBone link. At each receiver, we made two sets of measurements on this traffic stream using the `mtrace` and `mbat` software tools.

We used `mtrace` to determine the topology of the multicast tree. `mtrace` traces the *reverse* path from a multicast source to a receiver. It runs at the receiver and issues trace queries that travel hop by hop along the multicast tree towards the source. Each router along the path responds to these queries with its own IP address. We determined the tree topology by combining this path information for all receivers.

We also used `mtrace` to measure per-link packet losses. Routers also respond to `mtrace` queries with a count of how many packets they have seen directed to the specified multicast group. `mtrace` calculates packet losses on a link by comparing the packet counts returned by the two routers at either end of the link. We ran `mtrace` every two minutes during each one-hour experiment. These `mtrace` queries were also used to verify that the topology remained constant during each experiment.

It is important to note that `mtrace` does not scale to measurements of large multicast groups if used in parallel from all receivers as we describe here. Parallel `mtrace` queries converge as they travel up the tree. Enough such queries will overload routers and links with measurement traffic. We used `mtrace` in this way only to validate MINC on relatively small multicast groups.

We used `mbat` to collect traces of end-to-end packet losses. `mbat` runs at a receiver, subscribes to a specified multicast group, and records the sequence number and arrival time of each incoming packet. We ran `mbat` at each receiver for the duration of each hour-long experiment.

We then segmented the `mbat` traces into two-minute subtraces corresponding to the two-minute intervals on which we collected `mtrace` measurements. Finally, we ran our loss inference algorithm on each two-

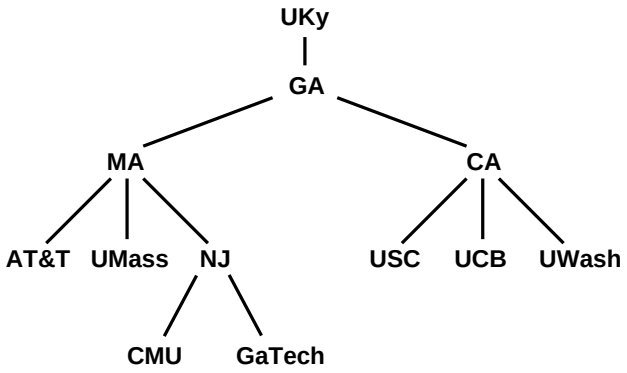


Figure 2: Multicast routing tree during our representative MBone experiment.

minute interval and compared the inferred loss rates with the directly measured loss rates.

Here we highlight results from a representative experiment on August 26, 1998. Figure 2 shows the multicast routing tree in effect during the experiment. The source was at the U. of Kentucky and the receivers were at AT&T Labs, U. of Massachusetts, Carnegie Mellon U., Georgia Tech, U. of Southern California, U. of California at Berkeley, and U. of Washington. The four branch routers were in California, Georgia, Massachusetts, and New Jersey.

Figure 3 shows that inferred and directly measured loss rates agreed closely despite a link experiencing a wide range of loss rates over the course of a one-hour experiment. Each short horizontal segment in the graph represents one two-minute, 1,200-probe measurement interval. As shown, loss rates on the link between the U. of Kentucky and Georgia varied between 4% and 30%. Nevertheless, differences between inferred and directly measured loss rates remained below 1.5%.

In summary, our MBone experiments showed that inferred and directly measured loss rates agreed closely under a variety of real network conditions:

- Across a wide range of loss rates (4%–30%) on the same link.
- Across links with very low ($< 1\%$) and very high ($> 30\%$) loss rates.
- Across all links in a multicast tree regardless of their position in the tree.
- Across different multicast trees.
- Across time of day and day of the week.

Furthermore, in all cases the inference algorithm converged to the desired loss rates well within each two-minute, 1,200-probe measurement interval.

A more detailed description of the experiments and results is found in [7], which is found in the Appendix.

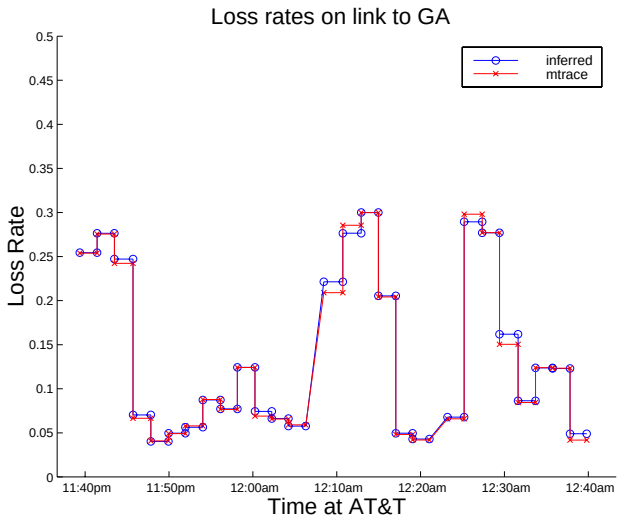


Figure 3: Inferred vs. actual loss rates on link between UKy and GA.

6.2 Simulation Experiments

We have performed more extensive validations of our inference techniques through simulation in two different settings: the simulation of the model with Bernoulli losses and simulations of networks with realistic traffic. In the model simulations, probe loss and delay obey the independence assumption of the model. We applied the inference algorithm to the end-to-end measurements and compared the inferred and actual model parameters for a large set of topologies and parameter values. We found that loss rates, mean delay, and variance estimates converged to close to their actual values with 2,000 probes. The number of probes required to accurately compute the entire delay distributions is higher. In our experiments we found good agreement with 10,000 probes.

The second type of experiment is based on the `ns` [20] simulator. Here delay and loss correspond to queueing delay and queue overflow at network nodes as multicast probes compete with traffic generated by TCP/UDP traffic sources. Multicast probes are generated by the source with fixed mean interarrival times; we used CBR or Poisson probes. We simulated different topologies with different background traffic mixes comprising infinite FTP sessions over TCP and exponential or Pareto on-off UDP sources. We considered both Drop Tail and Random Early Detection (RED) buffer discard methods, [17].

We compared the inferred loss and delay with actual probe loss and delay. We found rapid convergence of the estimates although with small persistent differences. We attribute this to the presence of spatial dependence, i.e., dependence between probe losses and delays on different links. This can arise through correlations in the background traffic due to correlation arising from TCP dynamics, e.g., synchronization between flows as a result of slow-start after packet loss. We have shown in [4] that small deviations from the spatial independence assumption lead to only small errors in inference.

We also found that background traffic introduces temporal dependence in probe behavior, e.g., its bursti-

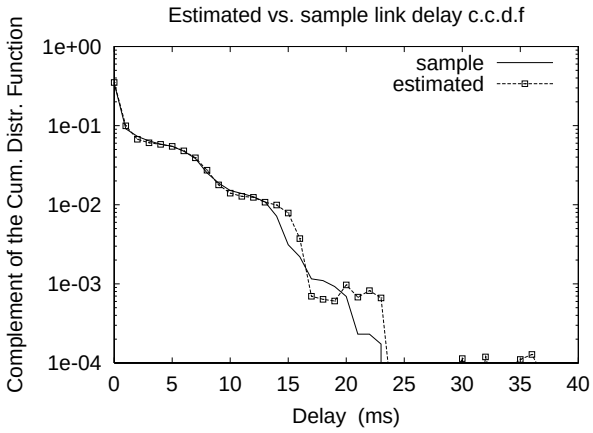


Figure 4: Inferred and Sample Delay ccdf. for a leaf link in the topology of Figure 2.

ness can cause back-to-back probe losses. We have shown that while temporal dependence can decrease the rate of convergence of the estimators, consistency is unaffected. In the experiments the inferred values converged within 2,000 probes despite the presence of temporal dependence.

While there is understanding of mechanisms by which temporal and spatial dependence can occur, as far as we know there are no experimental results concerning its magnitude. We believe that large or long lasting dependence is unlikely in the Internet because of traffic and link diversity. Moreover, we expect loss correlation to be reduced by the introduction of RED.

We also compared the inferred probe loss rates with the background loss rates. The experiments showed these to be quite close, although not as close as inferred and actual probe loss rates. We attribute this to the inherent difference in the statistical properties of probe traffic and background traffic.

To illustrate the distribution of delay inference results, we simulated the topology of the multicast routing tree shown in Figure 2. In order to capture the heterogeneity between edges and core of a network, interior links have higher capacity (5Mb/sec) and propagation delay (50ms) than those at the edge (1Mb/sec and 10ms). Background traffic comprises infinite FTP sessions and exponential on-off UDP sources. Each link is modeled as a FIFO queue with a 4-packet capacity. Real buffers are usually much larger; the capacity of four is used to reduce the time required to simulate the network. The discard policy is Drop Tail. In Figure 4, we plot the inferred versus the sample complementary cumulative distribution function (discretized in one millisecond bins) for one of the leaf links, using about 18,000 Poisson probes. The estimated distribution closely follows the sample distribution and is quite accurate for tail probabilities greater than 10^{-2}. Note that the estimated distribution is not always monotonically decreasing. This is because negative probabilities are occasionally estimated in the tail due to an insufficient number of samples. It is worth pointing out that, given the irregular shape of the sample distribution, the same level of accuracy would not be possible using a parametric model.

The evaluation of loss-based techniques can be found in [4] and [5], both of which are contained in the

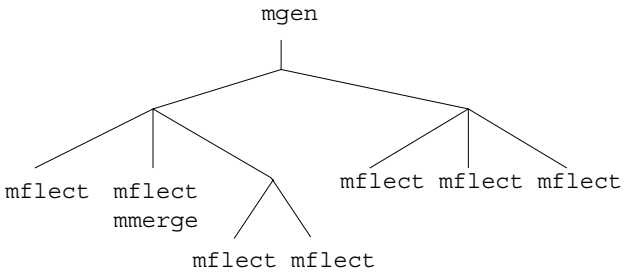


Figure 5: An RTCP-based tool deployment example, on the same topology as shown in Figure 2, with inference being performed at UMass.

Appendix. The evaluation of delay-based techniques can be found in [15], also contained in the Appendix.

7 Measurement and Analysis Tools

We have developed two sets of tools by which to deploy the MINC methodology. First, we have developed tools that leverage off of the real-time transport protocol, RTP, and its associated control protocol, RTCP, for generating and collecting end-to-end multicast measurement traces. Our RTP-based tool is described in Section 7.1. Second, we developed a web-based analysis and visualization tool, MINT (Multicast Inference Network Tool) for inferring loss behavior. This tool is described in Section 7.2.

7.1 Integration with RTCP

We have developed tools for applying MINC in real-time, so that MINC can be used by applications to respond to changing network conditions in new and more sophisticated ways. For example, a management program might adaptively adjust its probes to home in on a problem router.

Our tools transmit network information using RTCP, the control protocol for multicast transport protocol RTP [22]. By sharing their traces using RTCP, they benefit from RTCP's built-in scaling mechanisms.

The approach is based on three tools: `mgen`, `mflect`, and `mmerge` (Figure 5). `mgen` generates a stream of data (and may be replaced by any other application that multicasts data over RTP). A copy of `mflect` at each receiver maintains traces of the packets it does and does not receive from `mgen`. It periodically multicasts these (in a sense reflecting the data stream: hence "mflect"). `mmerge` collects the traces sent by `mflect`, collating those from the different data receivers and making them available to a tool, such as MINT, for inference.

`mflect` and `mmerge` are designed so that they may be incorporated directly into existing and future multicast applications. Their joint functionality is available as an extension to the RTP common code library from University College London, called RTPXR, ("eXtended Reporting"). An application using RTPXR would be in a position to respond adaptively to information on the topology of its data distribution tree.

In order that these tools scale to large numbers of receivers, we have developed *thinning* and *compression*

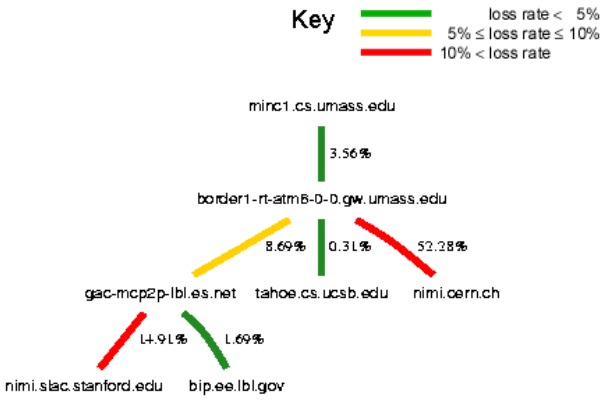


Figure 6: MINT view of the logical multicast tree with losses.

techniques. Thinning of receiver reports is necessary in order to avoid the abuse of bandwidth by receivers. We developed a *coordinated thinning* algorithm, which attempts to maximize the overlap between receiver reports, i.e., have receivers report on the same set of probes. Compression can further reduce bandwidth usage. Details of the approach can be found in [6], which is contained in the Appendix.

7.2 MINT: Multicast Inference Network Tool

MINT is intended to facilitate multicast-based inference. It takes as inputs all of the traces collected from the end-hosts. These traces may or may not include `mtrace` outputs. MINT comprises three components: a web-based user interface, a topology discovery algorithm, and an inference engine. Users interact with MINT to manipulate the inference such as choosing number of samples, visualizing the multicast tree with losses or showing the performance evolution over specific links. Depending on the availability of `mtrace` output, MINT discovers the topology by either parsing `mtrace` inputs or inferring the multicast tree from the loss traces. The inference engine takes topology information and loss traces to infer the network internal loss and then provides this to the user. The user can then view the results in one of several ways. One way is to lay out the logical multicast tree and display the links in different colors to distinguish different average loss rates (e.g., Figure 6). The user can also focus on a single link and observe how the loss rate evolves over time for that link.

8 Conclusions

We have presented the results obtained from our research on the use of end-end measurements to infer internal network behavior. We aslo described several tools obtained from this effort for obtaining measurements and for analyzing and visualizing the results from this analysis. All of the algorithms and their detailed evaluation can be found in the Appendix. Last, it is worth noting that this project has had considerable impact on research in this area. It constitutes the *first* effort to develop a rigorous foundation on top of which to de-

sign and evaluate end-to-end based network management tools. Many efforts are now under way exploring various extensions to the approaches developed under MINC. These include efforts at Rice, Boston Univ., UPenn, Lucent Bell Labs, Technion among others.

9 Bibliography

References

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley. "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior," *IEEE Communications Magazine*, May 2000.

[2] M. Adler, T. Bu, R. K. Sitaraman, D. Towsley. "Tree Layout for Internal Network Characterizations in Multicast Networks", *Proc. 3rd Intl Workshop on Networked Group Communication*, Nov. 2001. Also available as UMass CMPSCI Technical Report 00-44.

[3] T. Bu, N. Duffield, F. Lo Presti, D. Towsley. "Network Tomography on General Topologies," To appear in *Proc. SIGMETRICS 2002*.

[4] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, "Multicast-Based Inference of Network-Internal Loss Characteristics," *IEEE Transactions on Information Theory*, Nov. 1999.

[5] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," *Proc. IEEE Infocom '99*, March 1999.

[6] R. Caceres, N.G. Duffield, T. Friedman. "Impromptu Measurement Infrastructures using RTP". To appear in *Proc. IEEE INFOCOM'02*, June 2002.

[7] R. Cáceres, N.G. Duffield, S. B. Moon, D. Towsley. "Inference of Internal Loss Rates in the MBone," *Proc. IEEE/ISOC Global Internet '99*, December 1999.

[8] N.G. Duffield, J. Horowitz, F. Lo Presti. "Adaptive multicast topology inference," *Proc. IEEE Infocom 2001*, April 2001.

[9] N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley. "Multicast Topology Inference from End-to-end Measurements," *Proc. IP Traffic Measurement, Modeling and Management*, Sept. 2000.

[10] N.G. Duffield, J. Horowitz, F. Lo Presti, D.Towsley. "Network DelayTomography from End-to-end Unicast Measurements," *Proc. of the 2001 International Workshop on Digital Communications 2001-Evolutionary Trends of the Internet*, Taormina, Italy, Sept 17-20, 2001.

[11] N.G. Duffield, J. Horowitz, F. Lo Presti, D.Towsley. “Multicast Topology Inference from Measured End-to-End Loss, *IEEE Trans. on Information Theory*, 2002.

[12] N.G. Duffield, J. Horowitz, D. Towsley, W. Wei, T. Friedman. “Multicast-Based Loss Inference with Missing Data”, To appear in *IEEE Journal on Selected Areas in Communications*, 2002.

[13] N.G. Duffield, F. Lo Presti, “Multicast Inference of Packet Delay Variance at Interior Networks Links”, *Proc. Infocom’2000*, Tel Aviv, Israel, March 26–30, 2000.

[14] N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley. “Inferring link loss using striped unicast probes,” *Proc. IEEE Infocom 2001*, April 2001.

[15] F. Lo Presti, N.G. Duffield, J, Horowitz, D. Towsley. “Multicast-Based Inference of Network-Internal Delay Distributions,” UMass CMPSCI Technical Report 99-55, 1999.

[16] G.L. MacLachlan, T. Krishnan. *The EM algorithm and extensions.* John Wiley, New York, 1997.

[17] S. Floyd, V. Jacobson. “Reandom Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Trans. on Networking*, 1(4):397–413, Aug. 1993.

[18] `mtrace`: Print multicast path from a source to a receiver. `ftp://ftp.parc.xerox.com/pub/net-research/ipmulti`

[19] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, “An Architecture for Large-Scale Internet Measurement,” *IEEE Communications*, 36(8):48–54, August 1998.

[20] `ns`: Network simulator. `http://www-mash.cs.berkeley.edu/ns`

[21] S. Ratnasamy, S. McCanne, “Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths Using End-to-End Measurements,” *Proc. IEEE Infocom ’99*, March 1999.

[22] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, January 1996.

Appendix A Papers and Reports

The following papers are included in the Appendix. They can also be accessed through the MINC web site at http::www-net.cs.umass.edu/minc

1. R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, "Multicast-Based Inference of Network-Internal Loss Characteristics," *IEEE Transactions on Information Theory*, Nov. 1999.
2. T. Bu, N. Duffield, F. Lo Presti, D. Towsley. "Network Tomography on General Topologies," To appear in *Proc. SIGMETRICS 2002*.
3. N.G. Duffield, J. Horowitz, D. Towsley, W. Wei, T. Friedman. "Multicast-Based Loss Inference with Missing Data", To appear in *IEEE Journal on Selected Areas in Communications*, 2002.
4. F. Lo Presti, N.G. Duffield, J, Horowitz, D. Towsley. "Multicast-Based Inference of Network-Internal Delay Distributions," UMass CMPSCI Technical Report 99-55, 1999.
5. N.G. Duffield, F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Networks Links", *Proc. Infocom'2000*, Tel Aviv, Israel, March 26–30, 2000.
6. N.G. Duffield, J. Horowitz, F. Lo Presti, D.Towsley. "Multicast Topology Inference from Measured End-to-End Loss, *IEEE Trans. on Information Theory*, 2002.
7. N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley. "Multicast Topology Inference from End-to-end Measurements," *Proc. IP Traffic Measurement, Modeling and Management*, Sept. 2000.
8. N.G. Duffield, J. Horowitz, F. Lo Presti. "Adaptive multicast topology inference," *Proc. IEEE Infocom 2001*, April 2001.
9. N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley. "Inferring link loss using striped unicast probes," *Proc. IEEE Infocom 2001*, April 2001.
10. M. Adler, T. Bu, R. K. Sitaraman, D. Towsley. "Tree Layout for Internal Network Characterizations in Multicast Networks", *Proc. 3rd Intl Workshop on Networked Group Communication*, Nov. 2001. Also available as UMass CMPSCI Technical Report 00-44.
11. R. Cáceres, N.G. Duffield, S. B. Moon, D. Towsley. "Inference of Internal Loss Rates in the MBone," *Proc. IEEE/ISOC Global Internet '99*, December 1999.
12. R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," *Proc. IEEE Infocom '99*, March 1999.

Multicast-Based Inference of Network-Internal Loss Characteristics*

R. Cáceres† N.G. Duffield‡ J. Horowitz§ D. Towsley¶

Abstract

Robust measurements of network dynamics are increasingly important to the design and operation of large internetworks like the Internet. However, administrative diversity makes it impractical to monitor every link on an end-to-end path. At the same time, it is difficult to determine the performance characteristics of individual links from end-to-end measurements of unicast traffic. In this paper, we introduce the use of end-to-end measurements of *multicast* traffic to infer network-internal characteristics. The bandwidth efficiency of multicast traffic makes it suitable for large-scale measurements of both end-to-end and internal network dynamics.

We develop a Maximum Likelihood Estimator for loss rates on internal links based on losses observed by multicast receivers. It exploits the inherent correlation between such observations to infer the performance of paths between branch points in the tree spanning a multicast source and its receivers. We derive its rate of convergence as the number of measurements increases, and we establish robustness with respect to certain generalizations of the underlying model. We validate these techniques through simulation and discuss possible extensions and applications of this work.

1 Introduction

Background and Motivation. Fundamental ingredients in the successful design, control and management of networks are mechanisms for accurately measuring their performance. Two approaches to evaluating network performance have been:

(i) Collecting statistics at internal nodes and using network management packages to generate link-level performance reports; and

(ii) Characterizing network performance based on end-to-end behavior of point-to-point traffic such as that generated by TCP or UDP.

A significant drawback of the first approach is that gaining access to a wide range of routers in an administratively diverse network can be difficult. Introducing new measurement mechanisms into the routers themselves is likewise difficult because it requires persuading large companies to alter their products. Also,

*This work was sponsored in part by the DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

†AT&T Labs–Research, Rm. B125, 180 Park Avenue, Florham Park, NJ 07932, USA; E-mail: ramon@research.att.com

‡AT&T Labs–Research, Rm. B139, 180 Park Avenue, Florham Park, NJ 07932, USA; E-mail: duffield@research.att.com

§Dept. of Math. & Statistics, University of Massachusetts Amherst, MA 01003-4515, USA; E-mail: joeh@math.umass.edu

¶Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA; E-mail: towsley@cs.umass.edu

the composition of many such small measurements to form a picture of end-to-end performance is not completely understood.

Regarding the second approach, there has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [1, 2, 14, 19, 21, 22]). A number of ongoing measurement infrastructure projects (Felix [5], IPMA [7], NIMI [13] and Surveyor [28]) aim to collect and analyze end-to-end measurements across a mesh of paths between a number of hosts. **pathchar** [10] is under evaluation as a tool for inferring link-level statistics from end-to-end point-to-point measurements. However, much work remains to be done in this area.

Contribution. In this paper, we consider the problem of characterizing link-level loss behavior within a network through end-to-end measurements. We present a new approach based on the measurement and analysis of the loss behavior of *multicast* probe traffic. The key to this approach is that multicast traffic introduces correlation in the end-to-end losses measured by receivers. This correlation can, in turn, be used to infer the loss behavior of the links within the multicast routing tree spanning the sender and receivers. This enables the identification of links with higher loss rates as candidates for the origin of the degradation of end-to-end performance.

Using this approach, we develop *maximum likelihood estimators* (MLEs) of the link loss rates within a multicast tree connecting the sender of the probes to a set of receivers. These estimates are, initially, derived under the assumption that link losses are described by independent Bernoulli losses, in which case the problem is that of estimating the link loss rates given the end-to-end losses for a series of n probes. We show that these estimates are strongly consistent (converge almost surely to the true loss rates). Moreover, the asymptotic normality property of MLEs allows us to derive an expression for their rate of convergence to the true rates as n increases.

We evaluate our approach for two-, four-, and eight-receiver populations through simulation in two settings. In the first type of experiment, link losses are described by time-invariant Bernoulli processes. Here we find rapid convergence of the estimates to their actual values as the number of probes increases. The second type of experiment is based on **ns** [18] simulations where losses are due to queue overflows as probe traffic competes with other traffic generated by infinite data sources that use the Transmission Control Protocol (TCP) [24]. In the two- and four- receiver topologies with few background connections we find fast convergence although there are persistent, if small, differences between the inferred and actual loss rates.

The cause of these differences is that losses in our simulated network display spatial dependence (i.e., dependence between links), which violates the Bernoulli assumption. We believe that large and long-lasting spatial dependence is unlikely in a real network such as the Internet because of its traffic and link diversity. This is supported by experiments with an eight-receiver topology with diverse background traffic in which we found closer agreement between inferred and actual loss rates. Furthermore, we believe that the intro-

duction of Random Early Detection (RED) [6] policies in Internet routers will help break such dependence.

The potential for both spatial and temporal dependence of loss motivates investigation into their effect. Our analysis shows that dependence introduces inference errors in a continuous manner: if the dependence small, the errors in the estimates are also small. Furthermore, the errors are a second order effect: in the special case of a binary tree with statistically identical dependent loss on sibling links, the Bernoulli MLE of the marginal loss rates are actually unaffected for interior links of the tree. More generally, the MLE will be insensitive to spatial dependence of loss within regions of similar loss characteristics. Furthermore, the analysis shows how prior knowledge of the likely magnitude of dependence–e.g. from independent network measurements–could be used to correct the Bernoulli MLE.

We note that interference from TCP sources introduces temporal dependence (i.e., dependence between different packets) that also violates the Bernoulli assumption. This dependence is apparent in our simulated network, where probe losses often occur back-to-back due to burstiness in the competing TCP streams. Such dependence has also been measured in the Internet, but rarely involves more than a few consecutive packets [1]. The consistency of the estimator does not require independence between probes; it is sufficient that the loss process be ergodic. This property holds, e.g., when the dependence between losses has sufficiently short range. However, the rate of convergence of the estimates to their true values will be slower. We quantify this for Markovian losses by applying the Central Limit Theorem for the occupation times of Markov processes. We use this approach to compare the efficacy of two sampling strategies in the presence of Markovian losses. In our experiments, inferred loss rates closely tracked actual losses rates despite the presence of temporal dependence.

The work presented in this paper assumes that the topology of the multicast tree is known in advance. We are presently developing algorithms to infer the multicast tree from the probe measurements themselves.

We envisage deploying inference engines as part of a measurement infrastructure comprising hosts exchanging probes in a WAN. Each host will act as the source of probes down a multicast tree to the others. A strong advantage of using multicast rather than unicast traffic is efficiency. N multicast servers produce a network load that grows at worst linearly as a function of N. On the other hand, the exchange of unicast probes can lead to local loads which grow as N^2, depending on the topology. We illustrate this in Figure 1. In this example, $2N$ servers exchange probes. For unicast probes, the load on central link grows as N^2; for multicast probes it grows only as $2N$.

Related Work. There are a number of measurement infrastructure projects in progress, all based on the exchange of unicast probes between hosts in the current Internet. Two of these, IPMA (Internet Performance Measurement and Analysis) [7] and Surveyor [28], focus on measuring loss and delay statistics; in the former between public Internet exchange points, in the latter between hosts deployed at sites participating in Internet 2. A third, Felix [5], is developing linear decomposition techniques to discover network topology,

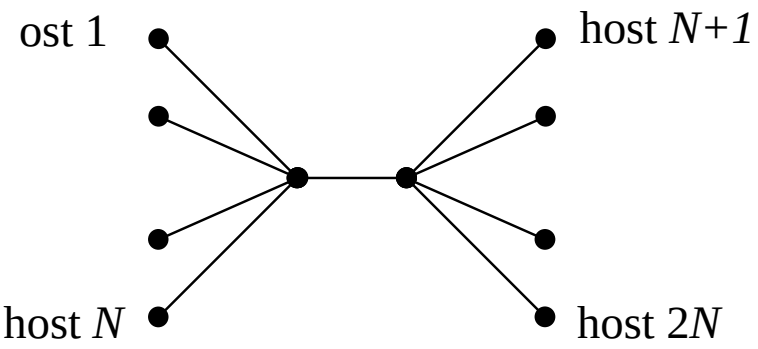


Figure 1: PROBE METHOD, LOAD AND TOPOLOGY: $2N$ servers exchange probes. For unicast probes, load on central link grows as N^2; for multicast probes it grows only as $2N$.

with an emphasis on network survivability. A fourth, NIMI (National Internet Measurement Infrastructure) [13], concentrates on building a general-purpose platform on which a variety of measurements can be carried out. These infrastructure efforts emphasize the growing importance of network measurements and help motivate our work. We believe our multicast-based techniques would be a valuable addition to these measurement platforms.

There is a multicast-based measurement tool, **mtrace** [16], already in use in the Internet. **mtrace** reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss and delay statistics. Topology discovery through **mtrace** is performed as part of the **tracer** tool [12].

However, **mtrace** suffers from performance and applicability problems in the context of large-scale measurements. First, **mtrace** traces the path from the source to a single receiver by working back through the multicast tree starting at that receiver. In order to cover the complete multicast tree, **mtrace** would need to run once for each receiver, which does not scale well to large numbers of receivers. In contrast, the inference techniques described in this paper cover the complete tree in a single pass. Second, **mtrace** relies on multicast routers to respond to explicit measurement queries. Current routers support these queries. However, Internet service providers may choose to disable this feature since it gives anyone access to detailed delay and loss information about paths in their part of the network. In contrast, our inference techniques do not rely on cooperation from any network-internal elements.

We now turn our attention to related theoretical work on inference methodologies. There has been some ad hoc, statistically non-rigorous work on deriving link-level loss behavior from end-to-end multicast measurements. An estimator proposed in [33] attributes the absence of a packet at a set of receivers to loss on the common path from the source. However, this is biased, even as the number of probes n goes to infinity.

For a different problem, some analytic methods for inference of traffic matrices have been proposed quite recently [30, 31]. The focus of these studies was to determine the intensities of individual source-destination flows from measurements of aggregate flows taken at a number of points in a network. Although

there are formal similarities in the inference problems with those of the present paper, the problem addressed by the other papers was substantially different. The solutions are not always unique or easily identifiable, sometimes needing supplementary methods to identify a candidate solution. This was a consequence of a combination of the coarseness of the data (average data rates in the class of Poissonian traffic processes) and the generality of the network topology considered.

Structure of the Paper. The remainder of the paper is structured as follows. In Section 2 we present a loss model for multicast trees and describe the framework within which analysis will occur. Section 3 contains the derivation of the estimators themselves; the specific example of the two-leaf tree is worked out explicitly. Section 4 analyzes the rates of convergence of estimators as the number of probes is increased. In particular, we obtain a simple approximation for estimator variance in the regime of small loss probabilities. In Section 5 we present an algorithm for computing packet loss estimates, and tests for consistency of the data with the model. Section 6 presents the results of simulation experiments that validate our approach. Motivated in part by the experimental results, we continue by examining the effects of violation of the Bernoulli assumption. In Section 7 we analyze the effects of spatial dependence on our estimators. We show how to correct for them on the basis of some a priori knowledge of their magnitude; we show that in any case they deform the estimates based on the Bernoulli assumption only to second order. In Section 8 we analyze the effect of temporal dependence on the loss process. We show that the asymptotic accuracy of the Bernoulli-based estimator is unaffected, although it may converge more slowly. We conclude in Section 9 with a summary of our contributions and proposals for further work. Some of the proofs are deferred to Section 10.

2 Model & Framework

2.1 Description of Logical Multicast Trees

Let $\mathcal{T} = (V, L)$ denote the logical multicast tree from a given source, consisting of the set of nodes V, including the source and receivers, and the set of links L. A link is ordered pair $(j, k) \in V \times V$ denoting a link from node j to node k. The set of children of a node j is denoted by $d(k)$ (i.e. $d(j) = \{k \in V : (j, k) \in L\}$). For each node $j \in V$ apart from the root 0, there is a unique node $k = f(j)$, the parent of j, such that $(j, k) \in L$. We shall define $f^n(k)$ recursively by $f^n(k) = f(f^{n-1}(k))$. We say that j is a descendant of k if $k = f^n(j)$ for some integer $n > 0$.

The root $0 \in V$ will represent the source of the probes. The set of leaf nodes $R \subset V$ (those with no children) will represent the set of receivers. The logical multicast tree has the property that every node has at least two descendants, apart from the root node (which has one) and the leaf-nodes (which have none). On the other hand, nodes in the full (as opposed to logical) multicast tree can have only one descendant. The logical multicast tree is obtained from the full multicast tree by deleting all nodes which have a single child (apart from the root 0) and adjusting the links accordingly. More precisely, if $i = f(j) = f^2(k)$ are

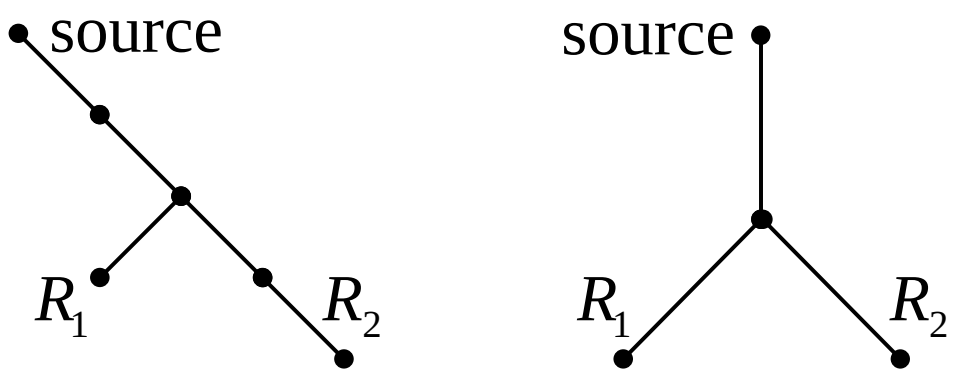


Figure 2: (a) A multicast tree with two receivers. (b) The corresponding logical multicast tree.

nodes in the full tree and $\#d(j) = 1$, then we assign to the logical tree only the nodes i, k and the link (i, k). Applying this rule to all such i, j and k in the full multicast tree yields the logical multicast tree.

A two receiver example is illustrated in Figure 2. A source multicasts a sequence of probes to two receivers, R_1 and R_2. The probes traverse the multicast tree illustrated in Figure 2(a). Figure 2(b) illustrates the logical multicast tree, where each path between branch points in the tree depicted in Figure 2(a) has been replaced by a single logical link.

2.2 Modeling the Loss of Probe Packets

We model the loss of probe packets on the logical multicast tree by a set of mutually independent Bernoulli processes, each operating on a different link. Losses are therefore independent for different links and different packets. In the introduction we discussed the reasons why network traffic can be expected to violate these assumptions; in Sections 7 and 8 we discuss the extent to which they affect the estimators described below, and how these effects can be corrected for.

We now describe the loss model in more detail. With each node $k \in V$ we associate a probability $\alpha_k \in [0, 1]$ that a given probe packet is not lost on the link terminating at k. We model the passage of probes down the tree by a stochastic process $X = (X_k)_{k \in V}$ where each X_k takes a value in $\{0, 1\}$; $X_k = 1$ signifies that a probe packet reaches node k, and 0 that it does not. The packets are generated at the source, so $X_0 = 1$. For all other $k \in V$, the value of X_k is determined as follows. If $X_k = 0$ then $X_j = 0$ for the children j of k (and hence for all descendants of k). If $X_k = 1$, then for j a child of k, $X_j = 1$ with independent probability α_j, and $X_j = 0$ with probability $\overline{\alpha}_j = 1 - \alpha_j$. (We shall write $1 - a$ as $\overline{a}$ in general). Although there is no link terminating at 0, we shall adopt the convention that $\alpha_0 = 1$, in order to avoid excluding the root link from expressions concerning the α_k. We display in Figures 3 and 4 examples of two- and four-leaf logical multicast trees which we shall use for analysis and experiments.

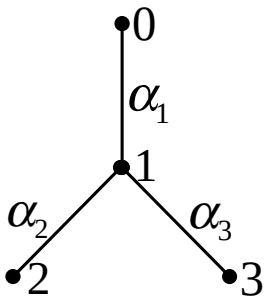


Figure 3: A two-leaf logical multicast tree

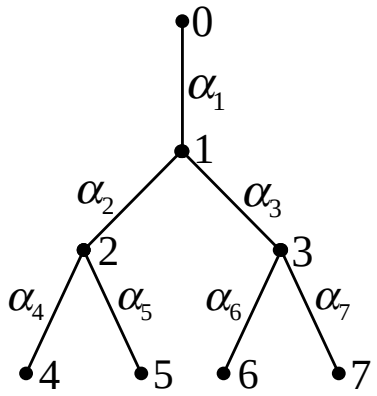


Figure 4: A four-leaf logical multicast tree

2.3 Data, Likelihood, and Inference

In an experiment, a set of probes is dispatched from the source. We can think of each probe as a trial, the outcome of which is a record of whether or not the probe was received at each receiver. Expressed in terms of the random process X, each such outcome is the set of values of X_k for k in the set of leaf nodes R, i.e. the random quantity $X_{(R)} = (X_k)_{k \in R}$, an element of the space $\Omega = \{0,1\}^R$ of all such outcomes. For a given set of link probabilities $\alpha = (\alpha_k)_{k \in V}$, the distribution of the outcomes $(X_k)_{k \in R}$ will be denoted by P_α. The probability mass function for a single outcome $x \in \Omega$ is $p(x;\alpha) = \mathsf{P}_\alpha(X_{(R)} = x)$.

Let us dispatch n probes, and, for each possible outcome $x \in \Omega$, let $n(x)$ denote the number of probes for which the outcome x obtained. The probability of n independent observations $x^1, \ldots, x^n$ (with each $x^m = (x_k^m)_{k \in R}$) is then

$$p(x^1, \ldots, x^n; \alpha) = \prod_{m=1}^{n} p(x^m; \alpha) = \prod_{x \in \Omega} p(x;\alpha)^{n(x)} \tag{1}$$

Our task is to estimate the value of α from a set of experimental data $(n(x))_{x \in \Omega}$. We focus on the class of *maximum likelihood estimators* (MLEs): i.e. we estimate α by the value $\breve{\alpha}$ which maximizes $p(x^1, \ldots, x^n; \alpha)$ for the data $x^1, \ldots, x^n$. Under very mild conditions, which are satisfied in the present situation, MLEs exhibit many desirable properties, including *strong consistency, asymptotic normality, asymptotic unbiasedness, and asymptotic efficiency* (see [11]). Strong consistency means that MLEs converge almost surely (i.e., with probability 1) to their target parameters as the sample size increases. The last three properties mean that, if the sample size is large, we can compute confidence intervals for the parameters at a given confidence level, the estimators are approximately unbiased, and there is no other estimator that would give the same level of precision with a smaller sample size.

Because of these properties, when a parametric model is available, MLEs are usually the estimators of choice. Moreover, the confidence intervals allow us to estimate the accuracy of the estimates of α, and in

particular their rate of convergence to the true parameter α as the number of samples n becomes large. This is important for understanding the number of probes which must be sent in order to obtain an estimate of α with some desired accuracy. Furthermore, in view of the possibility of large time-scale fluctuation in WANs, e.g. Internet routing instabilities as reported by Paxson [19], the period over which probes are sent should not be unnecessarily long.

3 The Analysis of the Maximum Likelihood Estimator

In this section we establish the form of the MLE and determine the rate at which it converges to the true value as the number of probes increases; this can be used to make prediction for given models, and also to estimate the likely accuracy of estimates derived from actual data. We work this out completely for the two-leaf tree of Figure 3.

3.1 The Likelihood Equation and its Solution

It is convenient to work with the log-likelihood function

$$\mathcal{L}(\alpha) = \log p(x^1, \ldots, x^n; \alpha) = \sum_{x \in \Omega} n(x) \log p(x; \alpha), \tag{2}$$

In the notation we suppress the dependence of $\mathcal{L}$ on n and $x^1, \ldots, x^n$. Since $\log$ is increasing, maximizing $p(x^1, \ldots, x^n; \alpha)$ is equivalent to maximizing $\mathcal{L}(\alpha)$.

We introduce the notation that $k \preceq k'$ for $k, k' \in V$ whenever k is a descendant of k' or $k = k'$ and $k \prec k'$ whenever $k \preceq k'$ but $k \neq k'$. We shall say that a link k is at level $\ell = \ell(k)$ if there is a chain of ℓ ancestors $k = f^0(k) \prec f(k) \prec f^2(k) \ldots \prec f^\ell(k) = 0$ leading back to the root 0 of $\mathcal{T}$. Levels 0 and 1 have only one node. We will occasionally use U to denote $V \setminus \{0\}$. Let $\mathcal{T}(k) = (V(k), L(k))$ denote the subtree within $\mathcal{T}$ rooted at node k. $R(k) = R \cap V(k)$ will be the set of receivers which are descended from k. Let $\Omega(k)$ be the set of outcomes x in which at least one receiver in $R(k)$ receives a packet, i.e.,

$$\Omega(k) = \{x \in \Omega : \bigvee_{j \in R(k)} x_j = 1\}. \tag{3}$$

Set $\gamma_k = \gamma_k(\alpha) = \mathsf{P}_\alpha[\Omega(k)]$. An estimate of γ_k is

$$\widehat{\gamma}_k = \sum_{x \in \Omega(k)} \widehat{p}(x), \quad \text{where} \quad \widehat{p}(x) := \frac{n(x)}{n}, \tag{4}$$

is the observed proportion of trials with outcome x. We will show that α can be calculated from $\gamma = (\gamma_k)_{k \in V}$, and that the MLE

$$\breve{\alpha} = \arg\max_{\alpha \in [0,1]^{\#R}} \mathcal{L}(\alpha) \tag{5}$$

can be calculated in the same manner from the estimates $\widehat{\gamma}$. The relation between α and γ is as follows. Define $\beta_k = \mathsf{P}[\Omega(k) \mid X_{f(k)} = 1]$. The β_k obey the recursion

$$\overline{\beta}_k = \overline{\alpha}_k + \alpha_k \prod_{j \in d(k)} \overline{\beta}_j, \quad k \in V \setminus R, \tag{6}$$

$$\beta_k = \alpha_k, \quad k \in R. \tag{7}$$

Then

$$\gamma_k = \beta_k \prod_{i=1}^{\ell(k)} \alpha_{f^i(k)}. \tag{8}$$

Theorem 1 *Let $\mathcal{A} = \{(\alpha_k)_{k \in U} : \alpha_k > 0\}$, and $\mathcal{G} = \{(\gamma_k)_{k \in U} : \gamma_k > 0 \, \forall k;\ \gamma_k < \sum_{j \in d(k)} \gamma_j \, \forall k \in U \setminus R\}$. There is a bijection Γ from $\mathcal{A}$ to $\mathcal{G}$. Moreover, Γ and Γ^{-1} are continuously differentiable.*

The proof of Theorem 1 relies of the following Lemma whose proof is given in Section 10.

Lemma 1 *Let C be the set of $c = (c_i)_{i=1,2,\ldots,i_{\max}}$ with $c_i \in (0,1)$ and $\sum_i c_i > 1$. The equation $(1-x) = \prod_i (1 - c_i x)$ has a unique solution $x(c) \in (0,1)$. Moreover, $x(c)$ is continuously differentiable on C.*

Proof of Theorem 1: The γ_k have been expressed as a function of the α_k, and clearly $\alpha_k > 0 \, \forall k \in U$ implies the conditions for $\mathcal{G}$. Thus it remains to show that the mapping from $\mathcal{A}$ to $\mathcal{G}$ is injective. Let $A_k = \prod_{i=0}^{\ell(k)} \alpha_{f^i(k)}$. From (8) we have

$$\gamma_k = A_k, \quad k \in R, \tag{9}$$

while combining (6) and (8) we find

$$H_k(A_k, \gamma) := (1 - \gamma_k/A_k) - \prod_{j \in d(k)} (1 - \gamma_j/A_k) = 0, \quad k \in U \setminus R. \tag{10}$$

Since $H_k(A_k, \gamma) = h(\gamma_k/A_k, \{\gamma_j/\gamma_k : j \in d(k)\})$ from Lemma 1, there is a unique $A_k > \gamma_k$ which solves (10). We recover the α_k uniquely from the A_k by taking the appropriate quotients (and setting $A_0 = \alpha_0 = 1$):

$$\alpha_k = A_k / A_{f(k)}, \quad k \in U. \tag{11}$$

Clearly Γ is continuously differentiable; that Γ^{-1} is also follows from the corresponding statement for $x(c)$ in Lemma 1. ■

Candidates for the MLE are solutions of the *likelihood equation* for the stationary points α of $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha) = 0, \quad k \in U. \tag{12}$$

Theorem 2 *When $\widehat{\gamma} \in \mathcal{G}$, the likelihood equation has the unique solution $\widehat{\alpha} := \Gamma^{-1}(\widehat{\gamma})$.*

Note that in the notation we have suppressed the dependence of $\breve{\alpha}$ and $\widehat{\alpha}$ on n and $x^1, \ldots, x^n$. We defer the proof of Theorem 2 to Section 10. That done, we must complete the argument by showing that the stationary point does have maximum likelihood. For this we must impose additional conditions. $\widehat{\alpha}$ is not precluded from being either a minimum or a saddle for the likelihood function, the maximum falling on the boundary of $[0,1]^{\#U}$. For some simple topologies we are able to establish directly that $\mathcal{L}(\alpha)$ is (jointly) concave in the parameters at $\alpha = \widehat{\alpha}$, which is hence the MLE $\breve{\alpha}$. For more general topologies we use an argument which establishes that $\widehat{\alpha} = \breve{\alpha}$ for all sufficiently large n, and whose proof also establishes some useful asymptotic properties of $\widehat{\alpha}$.

If $\alpha_k = 0$ for some link k, then $X_k = 0$ for all $j \in R(k)$, regardless of the values of α_j for j descended from k, and hence these cannot be determined. For this reason we now restrict attention to the case that all $\alpha_k > 0$, by passing to a subtree if necessary; see Section 5.

Theorem 3 *Assume $\alpha_k \in (0,1], k \in U$.*

(i) The model is identifiable*, i.e., $\alpha, \alpha' \in (0,1]^{\#R}$ and $\mathsf{P}_\alpha = \mathsf{P}_{\alpha'}$ implies $\alpha = \alpha'$.*

(ii) As $n \to \infty$, $\breve{\alpha} \to \alpha$ and $\widehat{\alpha} \to \alpha$, P_α almost surely.

(iii) Assume also $\alpha_k < 1$, $k \in U$. With probability 1, for sufficiently large n, $\breve{\alpha} = \widehat{\alpha}$.

Maximum Likelihood Estimator for the Two-leaf Tree Denote the 4 points of $\Omega = \{0,1\}^2$ by $\{00, 01, 10, 11\}$. Then

$$\widehat{\gamma}_1 = \widehat{p}(11) + \widehat{p}(10) + \widehat{p}(01), \quad \widehat{\gamma}_2 = \widehat{p}(11) + \widehat{p}(10), \quad \widehat{\gamma}_3 = \widehat{p}(11) + \widehat{p}(01). \tag{13}$$

The equations (10) for $\widehat{A}_k$ in terms of the $\widehat{\gamma}_k$ can be solved explicitly; combining with (11) we obtain the estimates

$$\widehat{\alpha}_1 = \frac{\widehat{\gamma}_2\widehat{\gamma}_3}{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1} = \frac{(\widehat{p}(01) + \widehat{p}(11))(\widehat{p}(10) + \widehat{p}(11))}{\widehat{p}(11)} \tag{14}$$

$$\widehat{\alpha}_2 = \frac{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1}{\widehat{\gamma}_3} = \frac{\widehat{p}(11)}{\widehat{p}(01) + \widehat{p}(11)} \tag{15}$$

$$\widehat{\alpha}_3 = \frac{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1}{\widehat{\gamma}_2} = \frac{\widehat{p}(11)}{\widehat{p}(10) + \widehat{p}(11)} \tag{16}$$

Note that although it is possible that $\widehat{\alpha}_1 > 1$ for some finite n, this will not happen when n is sufficiently large, due to Theorem 3(ii).

There is a simple interpretation of the estimates in (15) and (16). With the $\widehat{p}$'s replaced by their corresponding true probabilities p, (15) would give the probability of receiving a probe at node 1, given that it known to be received at node 2. For independent losses, this is just the marginal probability that the probe is received at node 1. We have found, however, the corresponding formulas when there are more than 2 sibling nodes do not allow such a direct interpretation.

4 Rates of Convergence of Loss Estimator

4.1 Large Sample Behavior of the Loss Estimator

In this section we examine in more detail the rate of convergence of $\widehat{\alpha}$ and the MLE $\breve{\alpha}$ to the true value α. We can apply some general results on the asymptotic properties of MLEs in order to show that $\sqrt{n}(\breve{\alpha} - \alpha)$ is asymptotically normally distributed as $n \to \infty$; some general properties of MLEs ensure that the same hold for $\sqrt{n}(\widehat{\alpha} - \alpha)$, and with the same asymptotic variance. We can use these results in two ways. First, for models of loss processes with typical parameters, we can estimate the number of probes required to obtain an estimate with a given accuracy. Secondly, we can estimate the likely accuracy of $\widehat{\alpha}$ from actual probe data and associate confidence intervals to the estimates.

The fundamental object controlling convergence rates of the MLE $\breve{\alpha}$ is the *Fisher Information Matrix* at α. This is defined for each $\alpha \in (0,1)^{\#U}$ as the $\#U$-dimensional real matrix $\mathcal{I}_{jk}(\alpha) := \mathsf{Cov}\left(\frac{\partial \mathcal{L}}{\partial \alpha_j}(\alpha), \frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha)\right)$. It is straightforward to verify that $\mathcal{L}$ satisfies conditions (see Section 2.3.1 of [27]) under which $\mathcal{I}$ is equal to the following more convenient expression which we will use in the sequel:

$$\mathcal{I}_{jk}(\alpha) = -\mathsf{E}\frac{\partial^2 \mathcal{L}}{\partial \alpha_j \partial \alpha_k}(\alpha) \tag{17}$$

On the other hand, a direct calculation of the asymptotic variance of $\widehat{\alpha}$ follows from the Central Limit Theorem. The random variables $\widehat{\gamma}$ are asymptotically Gaussian as $n \to \infty$ with

$$\sqrt{n}\,(\widehat{\gamma} - \gamma) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma), \tag{18}$$

where $\sigma_{jk} = \lim_{n\to\infty} n\,\mathsf{Cov}(\widehat{\gamma}_j, \widehat{\gamma}_k)$, for $j,k \in U$. Here $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution. Since by Theorem 1, Γ^{-1} is continuously differentiable on $\mathcal{G}$, then by the Delta method (see Chapter 7 of [27]) $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ is also asymptotically Gaussian, so establishing the first part of the following theorem. We note that the matrices ν and $\mathcal{I}^{-1}(\alpha)$ agree on the interior of the parameter space, but, as we shall see below, $\mathcal{I}(\alpha)$ may be singular on the boundary. Let $D_{ij}(\alpha) = \frac{\partial \Gamma_i^{-1}}{\partial \gamma_j}(\Gamma(\alpha))$ and D^T denotes the transpose.

Theorem 4 *(i) When $\alpha_k \in (0,1]$, $k \in U$, then as $n \to \infty$,*

$$\sqrt{n}\,(\widehat{\alpha} - \alpha) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \nu), \quad \text{where} \quad \nu = D(\alpha) \cdot \sigma \cdot D^T(\alpha). \tag{19}$$

(ii) When $\alpha_k \in (0,1)$, $k \in U$ then $\mathcal{I}(\alpha)$ is non-singular and $\mathcal{I}^{-1}(\alpha) = \nu$.

(iii) When $\alpha_k \in (0,1)$, $k \in U$, $\sqrt{n}\ (\breve{\alpha} - \alpha)$ converges in distribution as $n \to \infty$ to a $\#U$-dimensional Gaussian random variable with mean 0 and covariance matrix $\mathcal{I}^{-1}(\alpha)$.

Theorem 4 enables us to determine, for example, that asymptotically for large n, with probability $1 - \delta$, the $\widehat{\alpha}$ will lie between the points

$$\alpha_k \ \pm\ z_{\delta/2}\sqrt{\frac{\mathcal{I}_{kk}^{-1}(\alpha)}{n}}, \tag{20}$$

where $z_{\delta/2}$ denotes the number that cuts off an area $\delta/2$ in the right tail of the standard normal distribution. This is used for a confidence interval of level $1-\delta$. As we are interested in a 95% confidence interval for single link measurements, we take $z_{\delta/2} \approx 2$.

Confidence Intervals for Parameters. With slight modification, the same methodology can be used to obtain confidence intervals for the parameters $\widehat{\alpha}$ derived from measured data from n probes. Following [4] we use the *observed Fisher Information*:

$$\widehat{\mathcal{I}}_{jk}(\widehat{\alpha}) = -\frac{\partial^2 \mathcal{L}}{\partial \alpha_j \partial \alpha_k}(\widehat{\alpha}), \quad \text{where} \quad \widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma}). \tag{21}$$

Now, the proof of Theorem 2 (see particularly (57)) shows that the $\partial \mathcal{L}/\partial \alpha_k$ depend on the $n(x)$ only through the combinations $n\widehat{\gamma}_k$. Hence the same is true for the $\partial^2 \mathcal{L}/\partial \alpha_j \partial \alpha_k$. Since $\mathsf{P}_{\widehat{\alpha}}[\Omega(k)] = \Gamma(\Gamma^{-1}(\widehat{\gamma}))_k = \widehat{\gamma}_k$, we have $\widehat{\mathcal{I}}(\widehat{\alpha}) = \mathcal{I}(\widehat{\alpha})$.

We then use confidence intervals for $\widehat{\alpha}_k$ of the form

$$\widehat{\alpha}_k \ \pm \ z_{\delta/2} \sqrt{\frac{\mathcal{I}^{-1}_{kk}(\widehat{\alpha})}{n}}. \tag{22}$$

This allows us to find simultaneous confidence regions from the asymptotic distribution for α for a given tree. An issue for further study is to understand how the confidence intervals change as the tree grows.

Example: Confidence Intervals for the Two-leaf Tree An elementary calculation shows that the inverse of the Fisher information matrix governing the confidence intervals for models in (20) is

$$\mathcal{I}^{-1}(\alpha) = \begin{pmatrix} \frac{\alpha_1(\overline{\alpha}_3 - \alpha_2(1+\alpha_3(\alpha_1-2)))}{\alpha_2\alpha_3} & \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_3} & \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_2} \\ \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_3} & \frac{\overline{\alpha}_2\alpha_2}{\alpha_1\alpha_3} & \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_1} \\ \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_2} & \frac{-\overline{\alpha}_2\overline{\alpha}_3}{\alpha_1} & \frac{\overline{\alpha}_3\alpha_3}{\alpha_1\alpha_2} \end{pmatrix}. \tag{23}$$

Here, the order of the coordinates is $\alpha_1, \alpha_2, \alpha_3$. The inverse of the observed Fisher information governing the confidence intervals for data in (22) is obtained by inserting (14)–(16) into (23). We note that in this case $\mathcal{I}$ is singular at the boundaries $\alpha_2 = 1$ and $\alpha_3 = 1$.

4.2 Dependence of Loss Estimator Variance on Topology

The variance of $\widehat{\alpha}$ determines the number of probes which must be used in order to obtain an estimate of a given desired accuracy. Thus it is important to understand how the variance depends on the underlying topology. Growth of the variance with the size of the tree might preclude application of the estimator to large internetworks. Long timescale instability has been observed in the Internet [19]; if the timescale required for accurate measurements approaches that at which variability occurs, the estimator's requirement of stationarity would be violated. In this section we show that the asymptotic variance ν of $\widehat{\alpha}$ is independent of topology for loss ratios approaching zero.

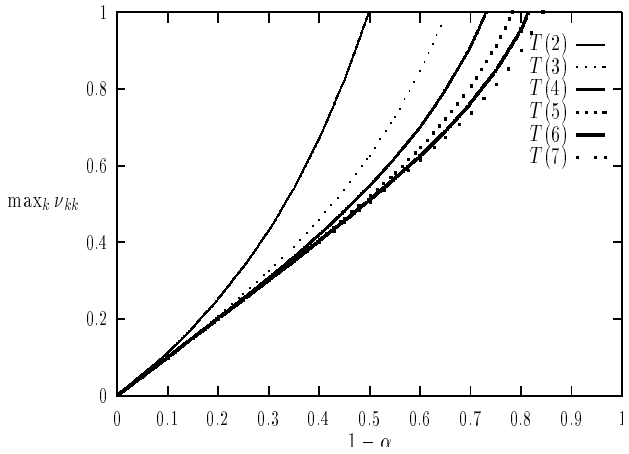


Figure 5: ASYMPTOTIC ESTIMATOR VARIANCE AND BRANCHING RATIO Depth 2 tree, 2 to 7 leaves. Variance decreases towards linear approximation $1-\alpha$ as branching ratio increases.

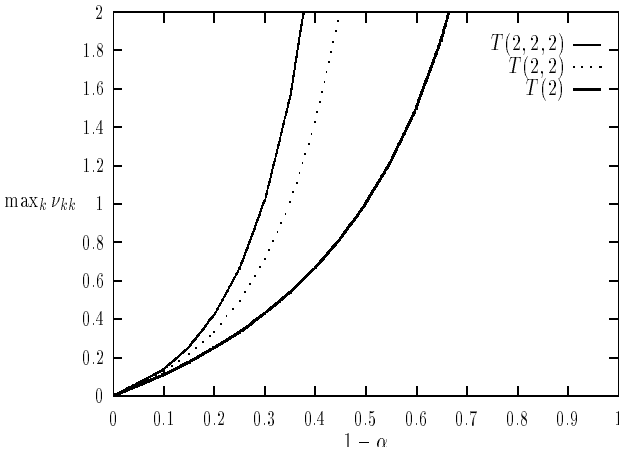


Figure 6: ASYMPTOTIC ESTIMATOR VARIANCE AND TREE DEPTH Binary Tree of depth 2, 3 or 4. Variance increases with tree depth.

The following theorem characterizes the behavior of ν for small loss ratio, independently of the topology of the logical tree. Set $\|\overline{\alpha}\| = \max_{k \in U} \overline{\alpha}_k$. Set $\delta_{jk} = 1$ if $j = k$ and 0 otherwise.

Theorem 5 $\nu_{jk} = \overline{\alpha}_k \delta_{jk} + O(\|\overline{\alpha}\|^2) \quad$ as $\|\overline{\alpha}\| \to 0$.

Theorem 5 says that the variance of $\widehat{\alpha}$ is, to first order in $\overline{\alpha}$, independent of topology. However, nothing is said about higher order dependence, and in particular whether the difference between ν_{jk} and $\overline{\alpha}_k \delta_{jk}$ converges to zero uniformly for all topologies as $\overline{\alpha} \to 0$. For a section of trees we used computer algebra to calculate the maximum asymptotic variance over links $\max_k \nu_{kk}$ for a selection of trees, as a function of the uniform Bernoulli probability $\alpha_k = \alpha$. We use the notation $T(r_1, r_2, \ldots, r_n)$ denote the tree of depth $n+1$ (depth = maximum level ℓ of any leaf) with successive branching ratios $1, r_1, r_2, \ldots, r_n$, i.e. the root node 0 has the single descendent node 1 which has r_1 descendents, each of which has r_2 descendents, and so on. We show the dependence on branching ratio in Figure 5 for trees of depth 2. In these examples, increasing the branching ratio decreases the variance. In Figure 6, we show the dependence on tree depth for binary trees of depth 2, 3 and 4. In this example, estimator variance increases with tree depth, roughly linearly. In all examples, estimator variance is approximately linear for $\overline{\alpha}$ less than about 0.1, and independent of topology, in keeping with Theorem 5. For larger $\overline{\alpha}$ it appears from these examples that the change in estimator variance of moving from simple topologies to more complex ones is governed by two opposing effects; variance reduction with increasing branching ratio, and variance growth with increasing tree depth. The reason for this appears to be that increasing the branching ratio increases the size of $R(k)$ (the set of leaf-nodes descended from k) so providing more data points for estimation, while increasing the tree depth increases cumulative error per link in estimation.

5 Data Consistency and Parameter Computation

In this section we address computational issues associated with the estimator $\widehat{\alpha}$. We specify consistency checks which must be applied to the data before $\widehat{\alpha}$ is computed. We describe an algorithm for computation of $\widehat{\alpha}$ and discuss its suitability for implementation in a network, in particular the extent to which it is distributable.

5.1 Data Consistency

In this section we describe tests for consistency of the empirical probabilities $\widehat{\gamma}$ with the model. The validations of the methodology carried out in this paper are all within controlled simulations. So we do not address here the additional consistency checks which would be required for applications to real network data, such as tests for stationarity.

The rest of this section focuses on range checking and tree surgery. An arbitrary data set $(n(x))_{x\in\Omega}$ may not give rise to $\widehat{\gamma} \in \Gamma((0,1)^{\#U})$. If this is because some of the $\widehat{\gamma}_k$ take values 0 or 1, then it can be dealt with by reducing the tree. In particular, when one of the $\widehat{\gamma}_k$ is 0, not all of the α_k can be inferred from the data. Those which cannot must be removed from consideration. In other cases, the data is not consistent with the assumptions that loss occurs independently on different links. We discuss these now.

(i) If $\widehat{\gamma}_k = 0$ for any $k \in V$, we construct a new tree by deleting node k and all its descendants, and perform the analysis on this pruned tree instead. We are unable to distinguish between the various ways in which γ_k may be zero, e.g. $\alpha_k = 0$, or $\alpha_k > 0$ but $\alpha_j = 0$ for children $j \in d(k)$.

(ii) If $\widehat{\alpha}_k = 1$ for any $k \in U$ then we can assign probability 1 to α_k. Then, for the purposes of calculation only, we consider a reduced tree obtained by excising node k in the same manner as nodes with a single descendant are excised from the physical multicast tree to generate the logical multicast tree; see Section 2.1.

(iii) Any $\widehat{\alpha}_k > 1$ is a nonphysical value, since the link probabilities are required to lie in $[0,1]$ (subject to (i) and (ii) above). Theorem 3 tells us this will not occur for sufficiently large n. Thus in implementations of the inference algorithm, this event may be used to trigger the dispatch of further probes.

(iv) The condition $\widehat{\gamma}_k = \sum_{j\in d(k)} \widehat{\gamma}_j$ for any $k \in U \setminus R$ prevents the calculation of $\widehat{A}_k$ and hence also link probabilities for links that include k as a vertex, namely $\widehat{\alpha}_k = \widehat{A}_k/\widehat{A}_{f(k)}$ and $\widehat{\alpha}_j = \widehat{A}_j/\widehat{A}_k$ for $j \in d(k)$. Instead, we estimate only the probabilities $\{\alpha_k\alpha_j : j \in d(k)\}$ on the composite links from $f(k)$ to the elements of $d(k)$, estimating $\widehat{\alpha_k\alpha_j} = A_j/A_{f(k)}, \quad j \in d(k)$. The possibility $\widehat{\gamma}_k > \sum_{j\in d(k)} \widehat{\gamma}_k$. is precluded by the relations (25) and (26) below. Equality occurs only if the observed losses satisfy the strong dependence property that each packet reaching a receiver in $R(k)$ reaches no other receiver in $R(k)$.

5.2 Computation of the Estimator on a General Tree

In this section we describe the algorithm for computing $\widehat{\alpha}$ on a general tree. An important feature of the calculation is that it can be performed recursively on trees. First we show how to calculate the $\widehat{\gamma}_k$. Denote by $(\widehat{X}_k(i))_{k\in R, i=1,2,..,n}$ the measured values at the leaf nodes of process X for n. Define the binary quantities $(\widehat{Y}_k(i))_{k\in V, i=1,2,..,n}$ recursively by

$$\widehat{Y}_k(i) = \widehat{X}_k(i), \quad k \in R \tag{24}$$

$$\widehat{Y}_k(i) = \bigvee_{j\in d(k)} \widehat{Y}_j(i), \quad k \in V \setminus R \tag{25}$$

so that

$$\widehat{\gamma}_k = n^{-1} \sum_{i=1}^{n} \widehat{Y}_k(i). \tag{26}$$

For simplicity we assume now that $\widehat{\gamma} \in \Gamma((0,1)^{\#U})$, so that, if necessary, steps (i) and (ii) of Section 5.1 have been performed on the data and/or the logical multicast tree in order to bring it to this form. The calculation of $\widehat{\alpha}$ can be done by another recursion. We formulate both recursions in pseudocode in Figure 7. The procedure **find_gamma** calculates the $\widehat{Y}_k$ and $\widehat{\gamma}_k$, assuming $\widehat{Y}_k$ initializes to $\widehat{X}_k$ for $k \in R$ and 0 otherwise. The procedure **infer** calculates the $\widehat{\alpha}_k$. The procedures could be combined. The full set of link probabilities is estimated by executing **main(**1**)** where node 1 is the single descendant of the root node 0.

Here, an empty product (which occurs when the first argument of **infer** is a leaf node) is understood to be zero. We assume the existence of a routine **solvefor** that returns the value of the first symbolic argument which solves the equation specified in its second argument. We know from Theorem 1 that under the conditions for $\widehat{\gamma}$ a unique such value exists.

5.3 Implementation of Inference in a Network

The recursive nature of the algorithm has important consequences for its implementation in a network setting. Observe that the calculation of $\widehat{\gamma}_k$ and A_k depends on X only through the $(\widehat{Y}_j)_{j\in d(k)}$. Put another way, if j is a child of k, the contribution to the calculation of $\widehat{\alpha}_k$ of all data measured at the set of receivers $R(j)$ descended from j, is summarized through $\widehat{Y}_j$. In a networked implementation this would enable the calculation to be localized in subtrees at a representative node, the computational effort at each node being at worst proportional to the depth of the tree (for the node that is the representative for all distinct subtrees to which it belongs).

Moreover, estimates from measurements at receivers $R(k)$ descended from a node k are consistent with those from the full set of receivers in the following sense. Executing **main(**k**)** yields the A_k calculated by **main(**1**)** as the value for $\widehat{\alpha}_k$. Thus is the effective probability that a probe traverse a (fictitious) link from the root 0 directly to k. But when the full inference **main(**1**)** is performed, it is not hard to see that the $\widehat{\alpha}$ obey $A_k = \prod_{i=0}^{\ell(k)} \widehat{\alpha}_{f^i(k)}$, i.e the probability of traversing the path from 0 to k without loss.

```
procedure main (k ) {
        find_gamma ( k ) ;
        infer ( k, 1 ) ;
}

procedure find_gamma ( k ) {
        foreach ( j ∈ d(k) ) {
                Ŷ_j = find_gamma ( j ) ;
                foreach ( i ∈ {1, ..., n} ) {
                        Ŷ_k[i] = Ŷ_k[i] ∨ Ŷ_j[i] ;
                }
        }
        γ̂_k = n^{-1} Σ_{i=1}^{n} Ŷ_k[i] ;
        return Ŷ_k ;
}

procedure infer ( k, A ) ;
        A_k = solvefor( A_k , (1 − γ̂_k/A_k) == ∏_{j∈d(k)} (1 − γ̂_j/A_k) );
        α̂_k = A_k/A ;
        foreach ( j ∈ d(k) ) {
                infer ( j , A_k ) ;
        }
}
```

Figure 7: PSEUDOCODE FOR INFERENCE OF LINK PROBABILITIES

6 Simulation Results

We evaluated our inference techniques through simulation and verified that they performed as expected. This work had two parts: *model simulations* and *TCP simulations*. In the model simulations, losses were determined by time-invariant Bernoulli processes. These losses follow the model on which we based our earlier analysis. In the TCP simulations, losses were due to queue overflows as multicast probes competed with other traffic generated by infinite TCP sources. We used TCP because it is the dominant transport protocol in the Internet [29]. The following two subsections describe our results from these two simulation efforts.

6.1 Model Simulations

Topology. For the model simulations, we used ad hoc software written in C++. We simulated the two tree topologies shown in Figures 3 and 4. Node 0 sent a sequence of multicast probes to the leaves. Each link exhibited packet losses with temporal and spatial independence. We could configure each link with a different loss probability that held constant for the duration of a simulation run. We fed the losses observed by the leaves to a separate Perl script that implements the inference calculation described earlier.

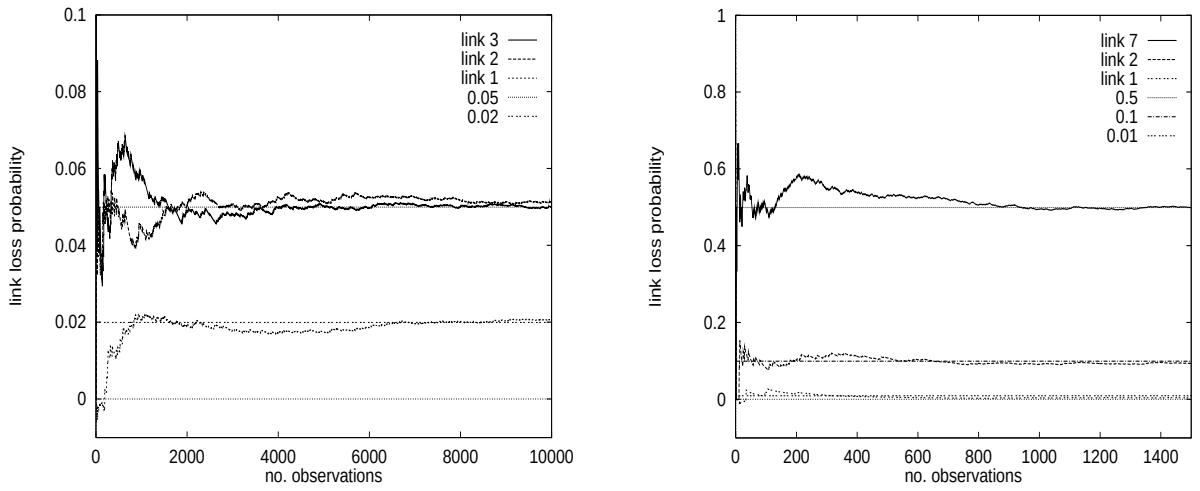


Figure 8: CONVERGENCE OF INFERRED LOSS PROBABILITIES TO ACTUAL LOSS PROBABILITIES IN MODEL SIMULATIONS. Left: Two-leaf tree of Figure 3 with parameters $\overline{\alpha}_1 = 0.02$; $\overline{\alpha}_2 = \overline{\alpha}_3 = 0.05$. Right: Selected links from four-leaf tree of Figure 4, with parameters $\overline{\alpha}_1 = 0.01$; $\overline{\alpha}_2 = 0.1$; $\overline{\alpha}_3 = \overline{\alpha}_4 = \overline{\alpha}_5 = \overline{\alpha}_6 = 0.01$; $\overline{\alpha}_7 = 0.5$. The graphs show that inferred probabilities converge to within 0.01 of the actual probabilities after 2,000 or fewer observations.

Convergence. Figure 8 compares inferred packet loss probabilities to actual loss probabilities. The left graph shows results for all three links in our two-leaf topology, while the right graph shows results for selected links in the four-leaf topology. In all cases, the inferred probabilities converge to within 0.01 of the actual probabilities after 2,000 observations.

Figure 9 compares the empirical and theoretical 95% confidence intervals of the inferred loss probabilities for the two-leaf topology. The empirical intervals were calculated over 100 simulation runs using 100 different seeds for the random number generator that underlies the Bernoulli processes. The theoretical intervals are as predicted by (20). As shown, simulation matches theory extremely well – we show the two graphs separately because the two sets of curves are indistinguishable when plotted together. For 2,000 observations, the confidence intervals lie with within 20% of the true probabilities.

It may seem that thousands of probes constitute too many network resources to expend and too long to wait for a measurement. However, it is important to note that a stream of 200-byte packets every 20 ms represents only 10 Kbps, equivalent to a single compressed audio transfer. Furthermore, a measurement using 5,000 such packets lasts less than two minutes. There already exist a number of MBone "radio" stations that send long-lived streams of sequenced multicast packets. In some cases we can use these existing multicast streams as measurement probes without additional cost. Overall, we feel that multicast-based inference is a practical and robust way to measure network dynamics.

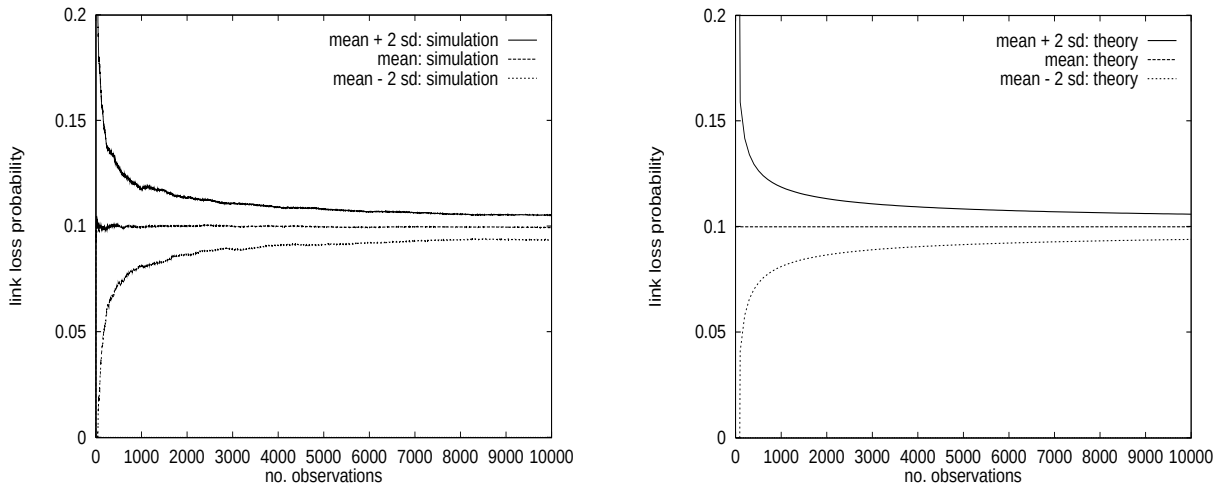


Figure 9: AGREEMENT BETWEEN SIMULATED AND THEORETICAL CONFIDENCE INTERVALS. Left: Results from 100 model simulations. Right: Predictions from (20). The graphs show two-sided confidence estimates at 2 standard deviations for link 2 of the four-leaf tree of Figure 4. Parameters were $\overline{\alpha}_1 = 0.01$; $\overline{\alpha}_2 = 0.1$; $\overline{\alpha}_3 = \overline{\alpha}_4 = \overline{\alpha}_5 = \overline{\alpha}_6 = 0.01$; $\overline{\alpha}_7 = 0.5$. Simulation matches theory extremely well – the two sets of curves are indistinguishable when plotted in the same graph.

6.2 TCP Simulations

Topology. For the TCP simulations, we used the **ns** network simulator [18]. We configured **ns** to simulate tree topologies shown in Figures 3, 4 and 11. All links had 1.5 Mbps of bandwidth, 10 ms of propagation delay, and were served by a FIFO queue with a 4-packet limit. Thus, a packet arriving at a link was dropped when it found four packets already queued at the link.

In each topology, node 0 sent multicast probe packets generated by a source with 200-byte packets and interpacket times chosen randomly between 2.5 and 7.5 msec. The leaf nodes received the multicast packets and monitored losses by looking for gaps in the sequence numbers of arriving probes. We fed the losses observed by the multicast receivers to the same inference implementation used for the model simulations described above. We also had **ns** report losses on individual links in order to compare inferred losses with actual losses.

In the two- and four-receiver topologies, each node maintained TCP connections to its child nodes. These connections used the Tahoe variant of TCP, sent 1,000-byte packets, and were driven by an infinite data source. Links to left children carried one such TCP stream, while links to right children carried two TCP streams. The link between nodes 0 and 1 also carried one TCP stream.

In the eight-receiver topology, the traffic more more diverse, with 52 TCP connections between different pairs of nodes, giving rise to approximately 8 connections per link on average.

Convergence. Figure 10 compares inferred loss rates to actual loss rates on selected links of our two- and

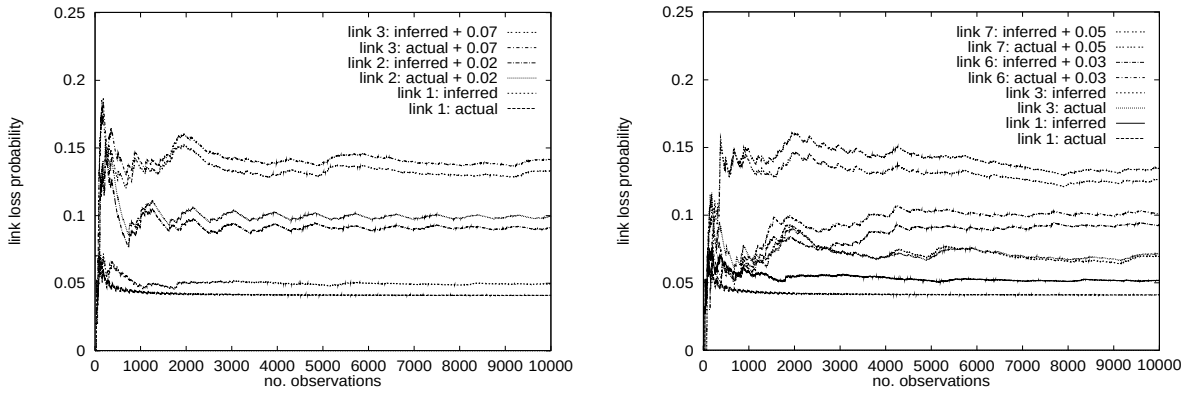


Figure 10: TRACKING OF ACTUAL LOSS RATES BY INFERRED LOSS RATES IN TCP SIMULATIONS. Left: Two-leaf tree of Figure 3. Right: Selected links from four-leaf tree of Figure 4 (some pairs of probabilities are offset for clarity). The graphs show that the inferred loss rates closely track the actual loss rates over 10,000 observations.

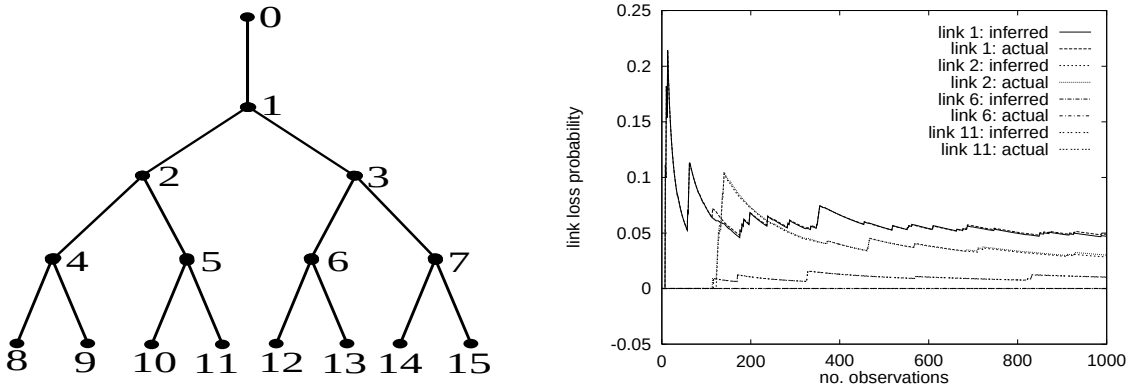


Figure 11: TRACKING OF ACTUAL LOSS RATES BY INFERRED LOSS RATES IN TCP SIMULATIONS WITH DIVERSE BACKGROUND TRAFFIC. LEFT: Eight-leaf binary tree. RIGHT: Close tracking of actual loss rates by estimated loss rates as number of observations is increased up to 1,000.

four-leaf topologies. As shown, the inferred rates closely track the actual rates over 10,000 observations. Figure 11 compares inferred and actual loss rates in the eight-receiver topology with diverse background traffic; in this case the tracking is even closer.

We note that the inferred values are accurate even though queue overflows due to TCP interference do not obey our temporal independence assumption. TCP is a bursty packet source, particularly in the region of exponential window growth during a slow start [9]. In our simulations, multicast probes are often lost in groups as they compete for queue space with TCP bursts. This phenomenon is readily apparent when watching animations of our simulations with the **nam** tool [17]. Inspection of the autocorrelation function of the time series of packet losses for a series of experiments predominantly showed correlation indistinguishable from zero beyond a lag of 1 (i.e. greater than back-to-back losses). As we explain in more

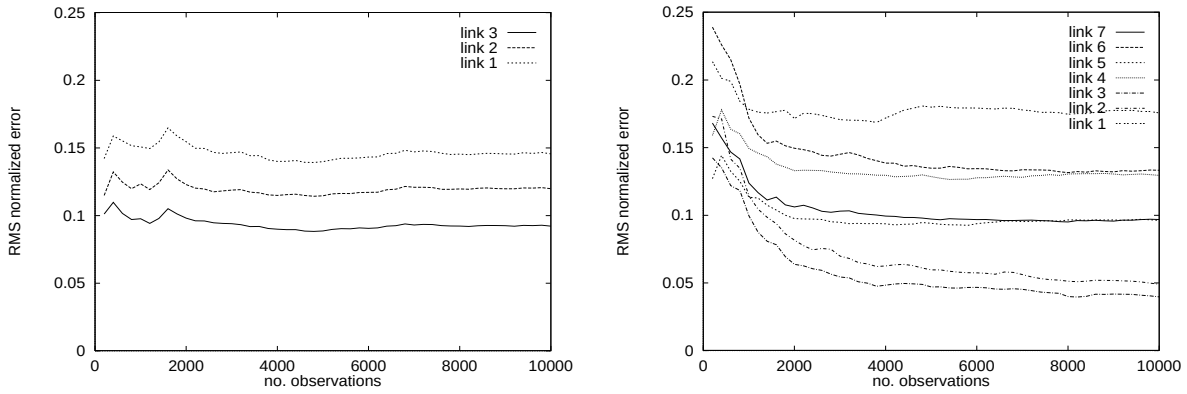


Figure 12: ACCURACY OF INFERENCE IN TCP SIMULATIONS. Left: Two-leaf tree of Figure 3. Right: Four-leaf tree of Figure 4. The graphs show normalized root mean square differences between actual and inferred loss rates, computed across 100 simulations. After an initial transient, inferred loss rates settle down to within 8 to 15% (in the two-leaf tree) and 4 to 18% (in the four-leaf tree) of actual loss rates, depending on the link. The RMS error was reduced to approximately 1% by modifying the MLEs to correct for spatial loss dependence.

detail in Section 8, the estimator $\widehat{\alpha}$ is still asymptotically accurate for large numbers of probes when losses have temporal dependence of sufficiently short range. However, the rate of convergence of the estimates to their true values will be slower.

Figure 12 shows the Root Mean Square (RMS) differences between the inferred and actual loss rates in the two- and four-leaf topologies. These differences were calculated over 100 simulation runs using 100 different seeds for the random number generator that governs the time between probe packets. As shown, the differences can drop significantly during the first 2,000 observations. However, at some point they level off and do not drop much further, if at all. This persistence reveals a systematic, although small, error in the inferred values because of spatial loss dependence. In our simulations, the same multicast probe is lost on sibling links more often than the spatial independence assumption dictates. These dependent losses lead the inference calculation to underestimate losses on the sibling links and to overestimate losses on the parent link.

We can quantify the spatial loss dependence present in the simulations. We can also calculate the effect of such dependence on the inferred loss probabilities by extending our previous analysis. Thus a prior estimate of the degree of dependence could be used to obtain corrections to the Bernoulli inference. We discuss this in more detail for spatial dependence in Section 7 and give an example of how to apply the correction. Applied to the inferences on the two-leaf tree summarized in Figure 10, they reduce an RMS error of between 8 and 15% to one of around 1%. The key observation behind these analyzes is that the error in the inferred values varies smoothly with the degree of spatial dependence. The greater the dependence in the network, the larger the error. We can arrange for correlated losses in a simulated network,

for example by creating synchronized interference streams on sibling links. However, the results for the eight-receiver topology with diverse background traffic support our belief that large and long-lasting spatial loss dependence is unlikely in real networks like the Internet because of their traffic and link diversity.

7 The Analysis and Correction of Spatial Dependence

7.1 Analysis of Spatial Dependence

When spatial dependence present in packet losses, the Bernoulli model assumption is violated. But even with such dependence, we can still ask what are the *marginal* loss probabilities for each link separately. In this section we quantify the effects of this dependence and show how they may be corrected for on the basis of a priori knowledge of them. We propose that this knowledge should be obtained by independent measurements on instrumented networks. Moreover, we establish that dependence deforms the Bernoulli estimates *continuously* in the sense that small divergences from independence of the losses lead to small divergence of the estimates of the marginal loss probabilities from their true values. For binary trees we find that the effect of such dependence on the estimates of marginal loss probabilities for links in the interior of the network is second order, and become negligible in regions of the network across which loss and dependence change little.

One motivation for considering dependent losses comes from the well-known example of synchronization between TCP flows which can occur as a result of the slow-start after packet loss; see [9]. Flows which have experienced common loss on a link k will then have some degree of dependence. Viewed as background traffic against which the probe packets compete, they can be expected to give rise to dependent losses of probe packets on links on the subtree descended from k. However, the dependence of probe loss can be expected to decrease on progressing down the tree from k. This happens if we assume that flows which became dependent though losses a given node k typically have a spread of destination address; then their paths through the network will subsequently diverge. Then the fraction of the total traffic contributed on links descended from k will decrease on progressing down the tree from k; hence the dependent influence of such flows on probe loss will decrease likewise.

The foregoing discussion motivates us to capture such dependence to first order by considering, within the class of dependent loss processes, those for which dependence only occurs between losses on sibling links, i.e., between those X_j and $X_{j'}$ for which $f(j) = f(j')$. Let $\Delta = \{\{j_1, \ldots j_n\} \subset d(k), k \in V \setminus R\}$ denote the set of subsets of sibling links. We characterize the joint distribution of the $(X_k)_{k\in V}$ through the family of joint conditional probabilities $(\alpha_{k_1,\ldots,k_n})_{\{j_1,\ldots j_n\}\in\Delta}$ where for $k = f(j_1) = \ldots = f(j_n)$,

$$\alpha_{j_1,\ldots,j_n} := \mathsf{P}[X_{j_1} = 1, \ldots, X_{j_n} = 1 | X_k = 1] \tag{27}$$

(For Bernoulli loss, $\alpha_{j_1,\ldots,j_n} = \prod_{m=1}^{n} \alpha_{j_m}$). We now derive analogous relations to (6) in this case. It is

convenient to work initially with the quantities

$$\xi_k := \mathsf{P}[\Omega(k) \mid X_k = 1] = \mathsf{P}[\Omega(k) \mid X_{f(k)} = 1]/\mathsf{P}[X_k = 1 \mid X_{f(k)} = 1] = \beta_k/\alpha_k \tag{28}$$

For $n \leq \#d(k)$ let $d_n(k)$ denote the set of subsets of $d(k)$ of cardinality n. By the Inclusion-Exclusion Principle (see e.g. Chapter 5.2 of [25])

$$\mathsf{P}[\Omega(k)] = \mathsf{P}\left[\cup_{j\in d(k)}\Omega(j)\right] = \sum_{n=1}^{\#d(k)} (-1)^{n+1} \sum_{\{j_1,\ldots,j_n\}\subset d_n(k)} \mathsf{P}\left[\Omega(j_1)\cap\ldots\cap\Omega(j_n)\right], \tag{29}$$

from which we find using (27) and (28) that

$$\xi_k = \sum_{n=1}^{\#d(k)} (-1)^{n+1} \sum_{\{j_1,\ldots,j_n\}\subset d_n(k)} \alpha_{j_1,\ldots,j_n}\xi_{j_1}\ldots\xi_{j_n} \tag{30}$$

Reexpressed in term of the γ_k we obtain the following analog of (10) for $k \in U \setminus R$:

$$H_k(A_k,\gamma,\psi) := \gamma_k/A_k - \sum_{n=1}^{\#d(k)} (-1)^{n+1} \sum_{\{j_1,\ldots,j_n\}\subset d_n(k)} \psi_{j_1,\ldots,j_n} \frac{\gamma_{j_1}\cdots\gamma_{j_n}}{A_k^n} = 0 \tag{31}$$

where $\psi_{j_1,\ldots,j_n} = \alpha_{j_1,\ldots,j_n}/(\alpha_{j_1}\ldots\alpha_{j_n})$ and we write $\psi = (\psi_{j_1,\ldots,j_n})_{\{j_1,\ldots,j_n\}\in\Delta}$. For a given loss model one can in principle compute ψ and compute A_k from γ_k. Rather than do this, however, we establish some structural results.

We can compare the actual values $A_k(\psi)$ which solve (31) for A_k, with those obtained from (10) with the Bernoulli assumption, which we can write as $A_k(1)$. The following theorem shows that the deformation from $A_k(\psi)$ to $A_k(1)$ is continuous in the neighborhood of the Bernoulli values $\psi = 1$ (i.e. $\psi_{j_1,\ldots,j_n} = 1$ for all $\{j_1,\ldots,j_n\} \in \Delta$).

Theorem 6 *Let $\alpha_k > 0$. There exists a neighborhood of $\psi = 1$ in $\mathbb{R}^{\#\Delta}$ on which $\psi \mapsto A_k(\psi)$ is continuous.*

Proof of Theorem 6: The result then follows from the Implicit Function Theorem (see [26]) provided that $\partial_{A_k} H_k(A_k(1),\gamma,1) \neq 0$. But $H_k(A_k,\gamma,1) = H_k(A_k,\gamma) = h(\gamma_k/A_k, \{\gamma_j/\gamma_k : j \in d(k)\})$ appearing in (10) and Lemma 1, and so the result follows from $\partial_x h(x(c),c) < 0$ as established during the proof of Lemma 1. ■

7.2 Spatially Dependent Losses in Binary Trees

When $\mathcal{T}$ is a binary tree we can obtain explicit results. For $k \in U\setminus R$ write $\psi^{(k)} = \psi_{jj'}$ where $d(k) = \{j, j'\}$ Then from (31) we have

$$\gamma_k = \begin{cases} A_k, & k \in R \\ \gamma_j + \gamma_{j'} + \psi^{(k)}\gamma_j\gamma_{j'}/A_k, & k \in U \setminus R \end{cases} \tag{32}$$

Let $\alpha(\psi)$ be the true value of α, i.e. that obtained by combining (32) with (11). $\alpha(1)$ is then the value previously obtained using the Bernoulli assumption. Let $k = 1$ denote the single descendent of the root node 0.

Theorem 7 *Let $\mathcal{T}$ be a binary tree.*

(i) There is a bijection Γ_ψ from $\mathcal{A}$ to $\mathcal{G}$ such that $\Gamma_\psi^{-1}(\gamma) = \alpha(\psi)$, with $\Gamma_1 = \Gamma$ from Theorem 1.

(ii)

$$\alpha_k(\psi) = \begin{cases} \alpha_1(1)/\psi^{(1)}, & k = 1 \\ \alpha_k(1)\psi^{(f(k))}, & k \in R \\ \alpha_k(1)\psi^{(f(k))}/\psi^{(k)}, & \text{otherwise} \end{cases} \tag{33}$$

Proof of Theorem 7: From (32), $A_k(\psi) = (\gamma_j + \gamma_{j'} - \gamma_k)/(\gamma_j\gamma_{j'}\psi^{(k)}) = A_k(1)/\psi^{(k)}$. The form of (ii) then follows from (11); this is used as the definition of Γ_ψ^{-1} for (i). ∎

Theorem 7(ii) has the interesting interpretation that in the interior of the network (i.e. except for node 1 and the leaf-nodes) the error in using $\alpha_k(\psi)$ in place of $\alpha_k(1)$ is a second order effect. For the error depends only on the on the relative magnitude of correlations at adjacent nodes through the quotient $\psi^{(f(k))}/\psi^{(k)}$. If the link probabilities and dependencies are (approximately) equal at each node of the tree, then this quotient will be (approximately) one, and so the Bernoulli estimate $\widehat{\alpha}_k(1) := \Gamma_1^{-1}(\widehat{\gamma})$ will be (approximately) equal to $\Gamma_\psi^{-1}(\widehat{\gamma})$, for interior k. Thus we see that the presence of dependent losses in binary trees perturbs the Bernoulli-based estimator little for links within the interior of regions across which the degree of dependence is similar. On the other hand, at the boundaries between such regions, a priori knowledge of the degree of dependence can help make the estimates more accurate. This motivates future work both in simulation studies and instrumentation of heterogeneous networks in order to establish the degree of dependence is influenced by dynamic factors such as utilization, and (comparatively) static factors such link technology and relative link speeds.

It is interesting to see that the TCP Simulations of the 4-leaf tree display some of the features one might expect from the above discussion. Observe in the RHS of Figure 10 that for the leaf-links (6 and 7) the inferred loss rate underestimates the actual loss rate, while for link 1 it overestimates it. For the interior link 3, the inferred and actual values are almost identical. This is consistent with the above discussion if $\psi_k > 1$ and $\psi_3 \approx \psi_{f(3)} = \psi_1$. Note that for $d(k) = \{j, j'\}$,

$$\psi_k > 1 \iff \alpha_{jj'} > \alpha_j\alpha_{j'} \iff \mathsf{E}[X_jX_{j'} \mid X_k = 1] > \mathsf{E}[X_j \mid X_k = 1]\mathsf{E}[X_{j'} \mid X_k = 1]. \tag{34}$$

In other words, $\psi_k > 1$ iff X_j and $X_{j'}$ are (conditional on $X_k = 1$) positively correlated. We expect this to be the case when synchronized losses occur as described at the start of this section.

	RMS difference from actual loss	
	adjusted	original
link 1	0.012	0.142
link 2	0.009	0.114
link 3	0.007	0.089

Table 1: CORRECTING FOR SPATIAL DEPENDENCE: RMS proportional difference of inferred from actual losses in **ns** simulation of two-leaf tree in Figure 3, after 10,000 probes. Adjustment of inference to account for dependence (left column) shows order of magnitude improvement over original inference (right column)

7.3 Correction for Spatial Dependence in Binary Trees

If some knowledge of the degree of dependence in the traffic is available, then this can be used to adjust the inferred loss probabilities accordingly. This motivates experimental studies of real networks with instrumented links in order to ascertain the magnitude of the dependence. We intend to undertake these experiments in the future. Here we show how knowledge of dependence can be used to correct the Bernoulli-based estimates of link probabilities for non-interior nodes. We consider the set of leaf-nodes $\{j, j'\} \in d(k)$. Let Y_j have the the distribution of X_j conditioned on $X_k = 1$. Suppose we know a priori an estimate $\widehat{\kappa}$ for the correlation of Y_j and $Y_{j'}$. Now the theoretical value of the correlation is

$$\kappa = \frac{\mathsf{Cov}(Y_j, Y_{j'})}{\sqrt{\mathsf{Var}(Y_j)\mathsf{Var}(Y_j)}} = \frac{\alpha_{jj'} - \alpha_j \alpha_{j'}}{\sqrt{\alpha_j \overline{\alpha}_j \alpha_{j'} \overline{\alpha}_{j'}}} = \psi^{(k)} \left(1 - \sqrt{\frac{\alpha_j \alpha_{j'}}{\overline{\alpha}_j \overline{\alpha}_{j'}}}\right) \tag{35}$$

Thus we expect to improve our estimates $\widehat{\alpha}_j(1)$ by using $\widehat{\alpha}_j(1)\widehat{\psi}^{(k)}$ instead where $\psi^{(k)}$ is obtained from (35) by using $\widehat{\kappa}$ and $\widehat{\alpha}(1)$ in place of κ and α.

To test this approach, we measured the loss dependence in an **ns** simulation of 10,000 probes in the two-leaf tree, then conducted 100 further **ns** simulations of 10,000 probes, and adjusted the inferred link probabilities in this manner. Comparing the actual, adjusted, and originally inferred loss ratios we see this provides improvement: the root mean square error goes down from between 8 and 15% (depending on the link) to about 1% in this case; see Table 1.

8 Temporal Dependence and Convergence Rates

8.1 Ergodicity and Asymptotic Accuracy

In this section we investigate the impact of temporal dependence on the estimator $\widehat{\alpha}$. Denote by $X(n) = (X_k(n))_{k \in V}$ the (spatial) process of the n^{th} probe. The first observation is that, if we replace the assumption of independence between probes to merely assuming that the (temporal) process $(X(n))_{n \in \mathbb{N}}$ is stationary and ergodic, then $\widehat{\alpha}$ still converges to α almost surely as the number of observations grows to ∞. This is because, by definition, the observed probabilities $\widehat{\gamma}$ of the ergodic process converge almost surely to the long

term averages. By stationarity, these are just the $\gamma = \Gamma(\alpha)$ as before, where the α are the (time)-marginal distributions of the link probabilities. A simple argument involving the Inverse Function Theorem (e.g., see [26]) shows that Γ^{-1} is continuous on $\Gamma((0,1)^{\#U})$, and hence $\widehat{\alpha} \to \alpha$ almost surely. Note we do not rely on $\widehat{\alpha}$ being the maximum likelihood estimator, with respect to some parameter space, for the marginal probabilities α of the general process. Rather, we have shown that the Bernoulli estimator is asymptotically accurate for stationary ergodic processes.

In the remainder of this section we examine the rate of convergence when X possesses temporal dependence. In an application of the method to measurement on real networks however, inherent variability (due do large scale events such as routing changes) may impose limits on the durations over which we can expect the loss process to be stationary. For this reason it is important to understand in more detail the impact of time-dependent packet loss on convergence rates. We propose to examine this through models. Markovian models of packet loss have been proposed on the basis of observations of the Internet (e.g., see [1]), although some longer bursts of losses were also found. We shall see that the price of temporal dependence is slower convergence than for the Bernoulli case. One can understand this qualitatively from the fact that burstiness in the packet loss processes means that the long-term average of $\widehat{\gamma}$ takes longer to approach.

8.2 Convergence Rates for Markovian Congestion

The main tool in understanding convergence rates is the following. Let Γ_k^{-1} denote the node k component of Γ^{-1}, so that $\widehat{\alpha}_k = \Gamma_k^{-1}(\widehat{\gamma})$. Suppose now that the random variables $\widehat{\gamma}$ are asymptotically Gaussian as $n \to \infty$ with

$$\sqrt{n}\,(\widehat{\gamma} - \gamma) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma), \tag{36}$$

where $\sigma_{jk} = \lim_{n\to\infty} n\,\mathsf{Cov}(\widehat{\gamma}_j, \widehat{\gamma}_k)$, for $j, k \in U$. Here $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution. Then by the Delta method (see Chapter 7 of [27]), since Γ_k^{-1} is continuously differentiable on $\mathcal{G}$ (see Theorem 1), $\Gamma_k^{-1}(\widehat{\gamma})$ is also asymptotically Gaussian:

$$\sqrt{n}\,(\Gamma_k^{-1}(\widehat{\gamma}) - \alpha_k) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \nu_k), \quad \text{where} \quad \nu_k = \nabla\Gamma_k^{-1}(\gamma) \cdot \sigma \cdot \nabla\Gamma_k^{-1}(\gamma). \tag{37}$$

In the remainder of this section we establish (36) within the context of Markov loss processes, and perform some explicit calculations for the 2-leaf tree.

We expand the class of loss processes as follows. We will define a Markov process $(Y(n))_{n\in\mathbb{N}}$, where $Y(n)$ will describe the state of the network encountered by the n^{th} probe; this description is used whether, for example, the interprobe times are constant, variable or random. Y is constructed as follows. For each $k \in U$ let $(Y_k(n))_{n\in\mathbb{N}}$ be an independent Markov process on the state space $\{0, 1\}$. We think of $Y_k(n)$ as representing the state of link k at time n, taking the value 0 if the link is congested, 1 if it is not. A probe that encounters a congested link is lost. We represent this by the process $X = (X_k(n))_{k\in U, n\in\mathbb{N}}$ defined by

letting $X_k(n)$ be conditionally independent of $(X_j(m), Y_j(m))_{k \prec j, m<n}$ given $(X_{f(k)}(n), Y_k(n))$, with

$$(X_k(n) \mid X_{f(k)}(n), Y_k(n)) = \begin{cases} 0, & X_{f(k)} = 0 \\ Y_k(n), & X_{f(k)} = 1 \end{cases} \tag{38}$$

When $Y_k(\cdot)$ is Bernoulli with probability α_k to be in the state 1, then the $X(n)$ are independent for each n, with the $X_k(n)$ distributed as described in Section 2.2. X is not a Markov process, but rather is a function of the Markov process Y. Moreover, $X(n)$ is a some function of $Y(n)$ alone, which we denote by χ. For each $k \in U$, let $\Phi(k)$ be the set of configurations y of Y such that $\chi(y)$ has outcome $\chi(y)_{(R)}$ in $\Omega(k)$, i.e.,

$$\Phi(k) = \{y \in \{0,1\}^{\#U} : \chi(y)_{(R)} \in \Omega(k)\}. \tag{39}$$

Let Q denote the transition matrix for Y, i.e., $Q = \otimes_{k \in U} Q(k)$ is the Kronecker product of the transition matrices of the individual Y_k. Let $q(k) = \{1 - \alpha_k, \alpha_k\}$ and let $q = \otimes_{k \in U} q(k)$ be the corresponding product distribution.

Theorem 8 *With the above notation, assume $\alpha_k \in (0,1)$ for all $k \in U$. Then (37) holds with*

$$\sigma_{jk} = \sum_{y \in \Phi(j)} \sum_{z \in \Phi(k)} \left[q_y(\delta_{yz} - q_z) + 2 \sum_{m=1}^{\infty} (Q^m_{yz} - q_y) q_z \right], \tag{40}$$

where Q^m denotes the m-step transition matrix.

Observe that in the Bernoulli case, the second term in (79) vanishes, while the first depends only on the marginal probabilities α. This means that the first term in (79) gives rise to the diagonal elements of (23); in what follows we can thus restrict our attention to the increase in the asymptotic variance as specified by the second term.

We parameterize the transition matrix of Y_k as

$$Q(k) = \begin{pmatrix} 1 - \alpha_k \overline{\omega}_k & \overline{\alpha}_k \overline{\omega}_k \\ \alpha_k \overline{\omega}_k & 1 - \overline{\alpha}_k \overline{\omega}_k \end{pmatrix}, \tag{41}$$

where $\overline{\omega}_k \in (0, 1/\max\{\alpha_k, \overline{\alpha}_k\}]$. ω_k parameterizes the burstiness of Y_k without changing its marginal probabilities. $Y_k(m)$ and $Y_k(m+1)$ are positively (or negatively) correlated when $\omega_k > 0$ (or $\omega_k < 0$). When $\omega_k = 0$, Y_k is Bernoulli. By calculation of the matrix powers of $Q(k)$ through its spectral decomposition, we find that $Q^n(k)_{yz} q_z(k)$ is given by the matrix

$$[Q^n(k)_{yz}] = \omega_k^n F(k) + G(k), \quad \text{where} \quad F(k) = \alpha_k \overline{\alpha}_k \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad G(k) = q(k) \otimes q(k). \tag{42}$$

Expanding $Q^n = \otimes_{k \in U} Q^n(k)$ and summing over n we find

$$\sum_{m=1}^{\infty} \left[(Q^m_{yz} - q_y) q_z \right] = \sum_{W \subset U} g(W) \left(\otimes_{k \in W} F(k) \right) \otimes \left(\otimes_{k \in U \setminus W} G(k) \right), \tag{43}$$

where $g(\emptyset) = 0$ and otherwise $g(W) = (\prod_{k \in W} \omega_k)/(1 - \prod_{k \in W} \omega_k)$.

8.3 Example: the Two-leaf Tree

Taking gradients in (14)–(16) and reexpressing them in terms of α we find

$$\nabla\Gamma_1^{-1}(\Gamma(\alpha)) = \frac{(1,-\overline{\alpha}_3,-\overline{\alpha}_2)}{\alpha_2\alpha_3}, \quad \nabla\Gamma_2^{-1}(\Gamma(\alpha)) = \frac{(-1,1,\overline{\alpha}_2)}{\alpha_1\alpha_3}, \quad \nabla\Gamma_3^{-1}(\Gamma(\alpha)) = \frac{(-1,\overline{\alpha}_3,1)}{\alpha_1\alpha_2}, \quad (44)$$

Using the notation (abc), with $a,b,c \in \{0,1\}$, to denote a value of $Y(n)$, we have from (13):

$$\Phi(1) = \{(111),(110),(101)\}, \quad \Phi(2) = \{(111),(110)\}, \quad \Phi(3) = \{(111),(101)\}. \quad (45)$$

For simplicity we set the α_k and ω_k equal to α,ω. Then (43) becomes

$$\begin{aligned}\sum_{m=1}^{\infty}\left[(Q_{yz}^m - q_y)q_z\right] &= \frac{\omega^3}{1-\omega^3}F(1)\otimes F(2)\otimes F(3) \quad (46)\\ &+\frac{\omega^2}{1-\omega^2}\left(F(1)\otimes F(2)\otimes G(3)+F(1)\otimes G(2)\otimes F(3)+G(1)\otimes F(2)\otimes F(3)\right)\\ &+\frac{\omega}{1-\omega}\left(F(1)\otimes G(2)\otimes G(3)+G(1)\otimes G(2)\otimes F(3)+G(1)\otimes F(2)\otimes G(3)\right).\end{aligned}$$

Combining (44), (45) and (46) in (37) in (46) with Theorem 8

$$\mathcal{I}_{11}^{-1} = \frac{\overline{\alpha}-\alpha(1+\alpha(\alpha-2))}{\alpha}, \quad (47)$$

$$\mathcal{I}_{22}^{-1} = \mathcal{I}_{33}^{-1} = \frac{\overline{\alpha}}{\alpha} \quad (48)$$

$$\nu_1 = \mathcal{I}_{11}^{-1} + \frac{\overline{\alpha}\omega(\alpha^2+\alpha\omega+\alpha^2\omega+\omega^2-\alpha\omega^2+2\alpha^2\omega^2+\omega^3-\alpha\omega^3+\alpha^2\omega^3)}{\alpha(1+\omega)(1-\omega^3)} \quad (49)$$

$$\nu_2 = \nu_3 = \mathcal{I}_{22}^{-1} + \frac{\overline{\alpha}\omega((\alpha+\omega)^2+\omega^2(\alpha^2+\omega))}{\alpha(1+\omega)(1-\omega^3)} \quad (50)$$

From (42), ω is the geometric decay rate of correlations. We can interpret $\tau = 1/(1-\omega)$ as the mean correlation time of the losses; $\tau = 1$ for Bernoulli losses. In Figure 13 we display the increase in asymptotic variance by plotting the ratio $\nu_1/\mathcal{I}_{11}^{-1}$ of the asymptotic variance with Markovian correlations to that without. We do this for $\alpha \in [.5,1]$ and $\tau \in [1,10]$. $\nu_2/\mathcal{I}_{22}^{-1}$ displayed very similar behavior. The ratio is increasing in correlation time τ, and in the link transmission probability α.

8.4 Temporal Dependence and Probing Methodology

An approach to avoiding the effect of temporal dependence would be to time probes at intervals larger than the typical correlation time of losses. Although this will reduce the number of probes required for a given level of convergence, the absolute time of convergence may increase due to the increased time between probes. Increasing the probes spacing by a factor τ', but with all probes lying within a given measurement period would increase the variance of the estimates by a factor τ' for independent losses. With Markovian losses, the effect of dependence between probes could be ameliorated by taking $\tau' > \tau$, the correlation

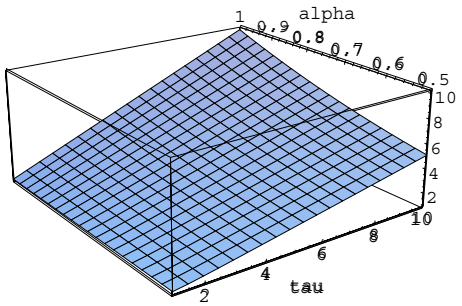


Figure 13: IMPACT OF TEMPORAL DEPENDENCE ON CONVERGENCE OF ESTIMATES: The ratio $\nu_1/\mathcal{I}_{11}^{-1}$ of the asymptotic variances of $\widehat{\alpha}_1$ with and without temporal dependence. Ratio is increasing in correlation time τ, and in link transmission probability α.

time. But for the two-leaf tree we see from (47) that when $\alpha \to 1$, then $\nu_k/\mathcal{I}_{kk}^{-1} \to 1/(1-\omega) = \tau$ for $k = 1, 2, 3$. Thus for small loss probabilities, the slow-down in the rate of convergence of $\widehat{\alpha}$ is no worse than that obtained by spacing probes to be approximately independent. In this example then, one may as well use all probes irrespective of their mutual dependence, rather than try to space them out to avoid dependence.

We envisage that direct measurement of the correlation time of received probes could be used, in combination with calculations of the previous section, to determine the number of probes, in an ongoing measurement, that are required in order to infer the link probabilities for a given accuracy. In the example considered we have seen that in order to estimate the increase in the asymptotic variation due to dependence between losses of small probability, it is sufficient to determine the correlation time of observed losses. When losses are heterogeneous, this will be conservative, since the autocorrelation will be dominated by the component with slowest decay.

A related issue is the randomization of interprobe times in order to avoid bias in the selection of network states which are observed via the probes. Probes with exponentially distributed spacings will see time averages; this is the PASTA property (Poisson Arrivals See Time Averages; see e.g. [32]). This approach has been proposed for network measurements [23] and is under consideration in the IP Performance Metrics working group of the IETF [8]. In the context of the above discussion, lengthening the interprobe time is to be understood as increasing the mean of the exponential distribution.

9 Summary and Future Work

In this paper, we introduced the use of end-to-end measurements of multicast traffic to infer network-internal characteristics. We developed statistically rigorous techniques for estimating packet loss rates on internal links, and validated these techniques through simulation. We showed that the inferred values quickly converged to within a small error of the actual values. We also presented evidence that our techniques yield accurate results even in the presence of moderate levels of temporal and spatial loss dependence.

We are extending our work in several directions. First, we are applying multicast-based inference to metrics other than packet loss. In particular, we have developed estimators for link delay. We are also investigating ways to infer link bandwidth and network topology using multicast probes. The ability to determine topology would free our measurements from the assumption of a priori knowledge of topology or of a separate topology-discovery tool.

Second, we plan to do more extensive simulations. We plan to substitute RED queueing for FIFO queueing to study the effect of RED on loss dependence. We also plan to substitute Poisson probes for CBR probes to avoid inadvertent synchronization of the probe traffic with periodic network processes. At the same time, we plan to simulate more complex topologies than the simple examples used throughout this paper. Topologies other than complete binary trees would stress our MLE for general trees, while larger topologies would test the convergence properties of our techniques on larger problem instances. This will be complemented by a theoretical analysis of the dependence of convergence rates on topology. Furthermore, we would like to explore how closely loss rates experienced by our probes agree with loss rates experienced by other network applications and protocols, for example TCP. We expect that our multicast-based measurements will yield ambient loss rates that are meaningful in a broad context.

Third, we plan to experiment with multicast-based inference on the Internet. As a preliminary step, we plan to measure ambient dependence in the real network, and determine the extent to which we need to adapt our estimates to their presence. We also plan to deploy our inference tools in multicast-enabled portions of the Internet, including the MBone, to test our techniques on a real network.

Finally, we would like to integrate our inference tools with one or more of the large-scale measurement infrastructures under construction. NIMI seems particularly suited because of its intended role as a general framework where many types of measurement can be carried out. The challenge will be to adapt a unicast-based infrastructure to perform multicast-based measurements, and in particular to schedule measurements, collect results, and perform inference calculations when large numbers of receivers are involved.

In conclusion, we feel that multicast-based inference is a powerful approach to measuring Internet dynamics. The rigorous statistical analysis behind our techniques gives them a firm theoretical footing, while the bandwidth efficiency of multicast traffic gives them much desired scalability. Robust and efficient measurements are increasingly important as the Internet continues to grow in size and diversity.

10 Proofs of Theorems

Proof of Lemma 1: Let $h_1(x) = (1-x)$, $h_2(x,c) = h_2(x) = \prod_i (1-c_i x)$. Let $q_i = c_i/(1-c_i x)$. Then for $x \in [0,1]$ $h_1''(x) = 0$, $h_2''(x) = h_2(x)\left\{(\sum_i q_i)^2 - \sum_i q_i^2\right\} > 0$. Hence $h(x) = h_1(x) - h_2(x)$ is strictly concave on $[0,1]$. Now $h(0) = 0$, $h(1) < 0$ and $h'(0) = -1 + \sum_i c_i > 0$. So since h is concave and continuous on $[0,1]$ there must be exactly one solution to $h(x) = 0$ for $x \in (0,1)$. Now set write

$h(x,c) = h_1(x) - h_2(x,c)$. Let $x(c)$ be the unique solution to $h(x(c),c) = 0$. The above derivation implies that $h'(x(c)) = (\partial h(x,c)/\partial x)|_{x=x(c)} < 0$, so in particular, is different from 0. Since h is continuously differentiable, then by the Implicit Function Theorem [26], so is $c \mapsto x(c)$. ∎

Proof of Theorem 2: The idea is to split up the sum (2) into portions on which $\frac{\partial \log p(x)}{\partial \alpha_k}$ is constant. These will be $\Omega(k)$, the $\Omega(f^i(k)) \setminus \Omega(f^{i-1}(k))$ for $i = 1, 2, \ldots, \ell(k)$, and $\Omega(0)^c$.

Consider first the case that $x \in \Omega(k)$. Then α_k occurs in $p(x)$ as a factor, and hence $\frac{\partial \log p(x)}{\partial \alpha_k} = 1/\alpha_k$. When $x \in \Omega(f^i(k)) \setminus \Omega(f^{i-1}(k))$ for $i = 1, 2, \ldots, \ell(k)$, then $p(x) = \overline{\beta}_{f^{i-1}(k)} R_k(x)$ where $R_k(x)$ does not depend on α_k (or indeed on any α_j for $j \leq f^{i-1}(k)$. Hence for $x \in \Omega(f^i(k)) \setminus \Omega(f^{i-1}(k))$,

$$\frac{\partial \log p(x)}{\partial \alpha_k} = \frac{1}{\overline{\beta}_{f^{i-1}(k)}} \frac{\partial \overline{\beta}_{f^{i-1}(k)}}{\partial \alpha_k} \tag{51}$$

Similarly, when $x \in \Omega(0)^c$,

$$\frac{\partial \log p(x)}{\partial \alpha_k} = \frac{1}{\overline{\beta}_0} \frac{\partial \overline{\beta}_0}{\partial \alpha_k} \tag{52}$$

On combining these:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \alpha_k} &= \frac{1}{\alpha_k} \sum_{x \in \Omega(k)} n(x) + \frac{1}{\overline{\beta}_0} \frac{\partial \overline{\beta}_0}{\partial \alpha_k} \sum_{x \in \Omega(0)^c} n(x) \\ &\quad + \sum_{i=1}^{\ell(k)} \left\{ \frac{1}{\overline{\beta}_{f^{i-1}(k)}} \frac{\partial \overline{\beta}_{f^{i-1}(k)}}{\partial \alpha_k} \sum_{x \in \Omega(f^i(k)) \setminus \Omega(f^{i-1}(k))} n(x) \right\} \end{aligned} \tag{53}$$

For the derivatives, some algebra with (7) shows that

$$\frac{\partial \overline{\beta}_k}{\partial \alpha_k} = -\beta_k/\alpha_k, \qquad \text{and} \tag{54}$$

$$\frac{\partial \overline{\beta}_{f^i(k)}}{\partial \alpha_k} = \frac{\alpha_{f^i(k)} - \beta_{f^i(k)}}{\overline{\beta}_{f^{i-1}(k)}} \frac{\partial \overline{\beta}_{f^{i-1}(k)}}{\partial \alpha_k} = -\frac{\beta_k}{\alpha_k} \prod_{m=1}^{i} \frac{\alpha_{f^m(k)} - \beta_{f^m(k)}}{\overline{\beta}_{f^{m-1}(k)}}. \tag{55}$$

The right hand term in equation (55) follows by iterating the middle term. Observe that

$$\sum_{x \in \Omega(f^i(k)) \setminus \Omega(f^{i-1}(k))} \frac{n(x)}{n} = \widehat{\gamma}_{f^i(k)} - \widehat{\gamma}_{f^{i-1}(k)} \quad \text{and} \quad \sum_{x \in \Omega(0)^c} \frac{n(x)}{n} = 1 - \widehat{\gamma}_0. \tag{56}$$

Combining (53), (54), (55) and (56) we get

$$\frac{\alpha_k}{n} \frac{\partial \mathcal{L}}{\partial \alpha_k} = \widehat{\gamma}_k - \beta_k \sum_{i=1}^{1+\ell(k)} \frac{\widehat{\gamma}_{f^i(k)} - \widehat{\gamma}_{f^{i-1}(k)}}{\overline{\beta}_{f^{i-1}(k)}} \prod_{m=1}^{i-1} \frac{\alpha_{f^m(k)} - \beta_{f^m(k)}}{\overline{\beta}_{f^{m-1}(k)}}. \tag{57}$$

Here we adopt the convention that the empty product for $i = 1$ means 1, and that the symbol $\widehat{\gamma}_{f(0)}$ that occurs when $i = 1 + \ell(k)$ means 1.

Set $\frac{\partial \mathcal{L}}{\partial \alpha_k}$ for all $k \in V$. For $k = 0$, (57) yields $0 = \widehat{\gamma}_0 - \beta_0(1 - \widehat{\gamma}_0)/\overline{\beta}_0$, whence

$$\widehat{\gamma}_0 = \beta_0 = \gamma_0. \tag{58}$$

For any other k, combining (57) for k and $j = f(k)$ yields

$$\widehat{\gamma}_k = \frac{\beta_k}{\overline{\beta}_k}\left(\widehat{\gamma}_j - \widehat{\gamma}_k + \frac{(\alpha_j - \beta_j)\widehat{\gamma}_j}{\beta_j}\right), \quad \text{whence} \quad \frac{\widehat{\gamma}_k}{\widehat{\gamma}_j} = \frac{\beta_k \alpha_j}{\beta_j} = \frac{\gamma_k}{\gamma_j}. \tag{59}$$

Together with (58) this gives $\widehat{\gamma}_k = \gamma_k$ for all $k \in V$. ∎

Proof of Theorem 3: (i) By the strong law of large numbers, $\widehat{\gamma} \to \Gamma(\alpha)$, P_α almost surely, as $n \to \infty$. Since Γ is, in particular, bijective, then the model is identifiable, since $\Gamma(\alpha) = \Gamma(\alpha')$ implies $\alpha = \alpha'$.

(ii) Convergence of $\widehat{\gamma}$ to γ (from (i)) and continuity of Γ^{-1} (from Theorem 1) yield convergence of $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ to $\alpha = \Gamma^{-1}(\gamma)$ as $n \to \infty$. We now establish convergence of $\breve{\alpha}$. Fix some $\alpha^0 \in (0,1)^{\#U}$, $M \subset (0,1)^{\#U}$, $x \in \Omega$ and define

$$Z(M, x) = \inf_{\alpha' \in M} \log \frac{p(x;\alpha^0)}{p(x;\alpha')} = \log p(x;\alpha^0) - \sup_{\alpha' \in M} \log p(x;\alpha'). \tag{60}$$

Observe that $p(x;\alpha)$ is polynomial in the α_k, and hence continuous. According to Lemma 7.54 in [27], it suffices to show that, for each $\alpha' \neq \alpha^0$, there is an open set $N_{\alpha'}$ containing α', such that $\mathsf{E}_{\alpha^0} Z(N_{\alpha'}, X) > -\infty$. (Here E_{α^0} is the expectation w.r.t. P_{α^0}).

Look at the two terms in $\mathsf{E}_{\alpha^0} Z(M, X)$ for any $M \subset (0,1)^{\#U}$. The first is $\mathsf{E}_{\alpha^0} \log p(X;\alpha^0) = \sum_{x\in\Omega} p(x;\alpha^0) \log p(x;\alpha^0)$. This is finite since $p \log p$ is bounded for $p \in [0,1]$ and Ω is finite. For the second term, note that $p(x;\alpha') \leq 1 \Rightarrow \log p(x;\alpha') \leq 0 \Rightarrow \sup_{\alpha'\in M} \log p(x;\alpha') \leq 0 \Rightarrow -\sup_{\alpha'\in M} \log p(x;\alpha') \geq 0 \Rightarrow \mathsf{E}_{\alpha^0} Z(M, X) \geq \mathsf{E}_{\alpha^0} \log p(X;\alpha^0) > -\infty$. Finally, we note that although it is not mentioned there, Lemma 7.54 in [27] requires identifiability, which we proved in (i) above.

(iii) Now let $\alpha \in (0,1)^{\#U}$ be the true set of link probabilities. From part (ii), with P_α probability 1, the MLE $\breve{\alpha} \to \alpha$ as $n \to \infty$. Hence, for each sequence of probes we have that for n sufficiently large, $\breve{\alpha}$ lies in the interior of $(0,1)^{\#U}$. For such n, $\breve{\alpha}$ must then solve the likelihood equation (12). We know from Theorem 2, that solutions of the likelihood equation are unique, and hence this $\breve{\alpha} = \widehat{\alpha}$. ∎

Proof of Theorem 4: (ii) Recall $V(k) = \{j \in V : j \preceq k\}$, $R(k) = V(k) \cap R$ and $U = V \setminus \{0\}$. Set $S(\alpha) = (S_k(\alpha))_{k\in U}$ with $S_k(\alpha) = \frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha)$ (the score vector). Then $\mathcal{I}_{jk}(\alpha) = \mathsf{Cov}(S_j(\alpha), S_k(\alpha)) = \mathsf{E}_\alpha(S_j(\alpha) S_k(\alpha))$ since $\mathsf{E}_\alpha(S_\alpha) = \sum_{x\in\Omega} p(x,\alpha) \frac{\partial}{\partial \alpha_k} \log p(x,\alpha) = \sum_{x\in\Omega} \frac{\partial}{\partial \alpha_k} p(x,\alpha) = 0$.

Suppose that $\mathcal{I}(\alpha)$ is singular for some $\alpha = (\alpha_k)_{k\in U} \in (0,1)^{\#U}$. Then there exists some nonzero vector $c = (c_k)_{k\in U}$ for which $c \cdot \mathcal{I} \cdot c = 0$. But $c \cdot \mathcal{I} \cdot c$ is the variance of the mean-zero random variable $c \cdot S(\alpha)$, so then we would have that $c \cdot \mathcal{S}(\alpha) = 0$, P_α almost surely, or equivalently

$$\sum_{k\in U} c_k \frac{\partial \log p(x,\alpha)}{\partial \alpha_k} = 0 \quad \forall x \in \Omega \tag{61}$$

since $\mathsf{P}_\alpha(\{x\}) > 0$ for all $x \in \Omega$. We show that, in fact, (61) implies $c_k = 0$, first for $k \in R$, then for all $k \in U$.

Let $x^{(0)} \in \Omega$ be such that $x_j^{(0)} = 1$ for all $j \in R$, and for some $k \in R$ let $x_j^{(1)} = 1$ for $j \neq k$ and 0 for $j = k$. Then

$$p(x^{(0)}, \alpha) = \prod_{j \in U} \alpha_j \quad \text{while} \quad p(x^{(1)}, \alpha) = \overline{\alpha}_k \prod_{j \in U \setminus \{k\}} \alpha_j \tag{62}$$

and so from (61)

$$\sum_{j \in U} \frac{c_j}{\alpha_j} = 0 \quad \text{while} \quad -\frac{c_k}{\overline{\alpha}_k} + \sum_{j \in U \setminus \{k\}} \frac{c_j}{\alpha_j} = 0. \tag{63}$$

Combining the last two equations we find $c_k = 0$.

We now proceed by induction. For $k \in U$ assume that $c_j = 0$ for all $j \prec k$. We now prove that $c_k = 0$. Let $x^{(0)}$ be as before, and set

$$x_j^{(3)} = \begin{cases} 1 & j \in R \setminus R(k) \\ 0 & j \in R(k). \end{cases} \tag{64}$$

Then

$$p(x^{(3)}, \alpha) = (\overline{\alpha}_k + \alpha_k \phi_k) \prod_{j \in V \setminus V(k)} \alpha_j \tag{65}$$

where $\phi_k = \prod_{j \in d(k)} \overline{\beta}_j = \mathsf{P}_\alpha[X_j = 0 \, \forall j \in R(k) \mid X_k = 1]$. Hence from (61)

$$\frac{c_k(\phi_k - 1)}{\overline{\alpha}_k + \phi_k \alpha_k} + \sum_{j \in V \setminus V(k)} \frac{c_j}{\alpha_j} = 0, \tag{66}$$

recalling the assumption that $c_j = 0$ for all $j \prec k$. For the same reason (61) reads

$$\frac{c_k}{\alpha_k} + \sum_{j \in V \setminus V(k)} \frac{c_j}{\alpha_j} = 0. \tag{67}$$

Combining (66) and (67), then we find $c_k = 0$. The equality of ν with $\mathcal{I}^{-1}$ in the interior of the space of parameters α is standard under the conditions established during the proof of Theorem 3; see, e.g., Chapter 6.4 of [11].

(iii) We refer to Theorem 7.63 of [27]. Clearly $\mathcal{L}$ is 3-times continuously differentiable on $(0,1)^{\#U}$, and has bounded expectation in some neighborhood of α. This establishes the relation (7.64) in [27]. $\frac{\partial \log p(x,\alpha)}{\partial \alpha_j \alpha_k}(\alpha)$ is clearly finite on $(0,1)^{\#U}$. Hence $\mathcal{I}$ is finite in $(0,1)^{\#U}$, so together with Theorem 3 and the non-singularity of $\mathcal{I}$ established in (ii) above, we are able to conclude the result. ∎

Proof of Theorem 5: Let $j \vee k$ denote the nearest common ancestor of j and k, i.e. $j \vee k$ is the $\prec$-least common upper bound of j and k. The proof proceeds by a number of subsidiary results. Since probes are assumed independent, it suffices to evaluate all random quantities for $n = 1$ probes.

(i) As $\|\overline{\alpha}\| \to 0$,

$$\text{(a) } 1 - A_k = s(k) + O(\|\overline{\alpha}\|^2); \quad \text{(b) } \overline{\beta}_k = O(\|\overline{\alpha}\|), \quad \text{(c) } 1 - \gamma_k = s(k) + O(\|\overline{\alpha}\|^2), \tag{68}$$

where

$$s(k) = \sum_{j \succeq k} \overline{\alpha}_j. \tag{69}$$

The relation (a) is clear by expanding $A_k = \prod_{j \succeq k}(1 - \overline{\alpha}_j)$. (b) follows by an inductive argument. Observe from (6) that if (b) holds for all $k \in d(j)$, it also holds for j. But since $\beta_k = \alpha_k$ for leaf-nodes $k \in R$, (b) holds for all k. (c) then follows from the relation $\gamma_k = A_k(1 - \prod_{j \in d(k)} \overline{\beta}_j)$.

(ii) As $\|\overline{\alpha}\| \to 0$,

$$\mathsf{Cov}(\widehat{\gamma}_j, \widehat{\gamma}_k) = s(j \vee k) + O(\|\overline{\alpha}\|^2) \tag{70}$$

To see this, we write $\mathsf{Cov}(\widehat{\gamma}_j, \widehat{\gamma}_k) = \mathsf{E}[\widehat{\gamma}_j\widehat{\gamma}_k] - \mathsf{E}[\widehat{\gamma}_j]\mathsf{E}[\widehat{\gamma}_k]$, and $\mathsf{E}[\widehat{\gamma}_j] = \gamma_j$ by definition. If k is an ancestor of j then $\widehat{\gamma}_j = 1 \Rightarrow \widehat{\gamma}_k = 1$ and so $\mathsf{E}[\widehat{\gamma}_j\widehat{\gamma}_k] = \gamma_j$. Similarly, if j is an ancestor of k, then $\mathsf{E}[\widehat{\gamma}_j\widehat{\gamma}_k] = \gamma_k$. Otherwise $\widehat{\gamma}_j = 1, \widehat{\gamma}_k = 1 \Rightarrow \widehat{\gamma}_{j\vee k} = 1$, and so we write $\mathsf{E}[\widehat{\gamma}_j\widehat{\gamma}_k] = \mathsf{P}[\widehat{\gamma}_j = 1 \mid X_{j\vee k} = 1]\mathsf{P}[\widehat{\gamma}_k = 1 \mid X_{j\vee k} = 1]\mathsf{P}[X_{j\vee k} = 1] = \mathsf{P}[\widehat{\gamma}_j = 1]\mathsf{P}[\widehat{\gamma}_k = 1]/\mathsf{P}[X_{j\vee k} = 1] = \gamma_j\gamma_k/A_{j\vee k}$. Thus,

$$\mathsf{Cov}(\widehat{\gamma}_j, \widehat{\gamma}_k) = \begin{cases} \gamma_k(1-\gamma_j) & j \succeq k \\ \gamma_j(1-\gamma_k) & k \succeq j \\ \gamma_j\gamma_k(1/A_{j\vee k} - 1) & \text{otherwise} \end{cases} \tag{71}$$

(70) then follows from (68) and the fact that $j \vee k = j$ when $j \succeq k$.

(iii) As $\|\overline{\alpha}\| \to 0$,

$$D(\alpha) = D + O(\|\overline{\alpha}\|) \quad \text{where} \quad D_{jk} := \begin{cases} 1 & k = j \\ -1 & k = f(j) \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

To establish this, note first that $D(\alpha)$ has inverse $D^{-1}(\alpha)$ whose elements are $(D(\alpha)^{-1})_{ij} = \partial\gamma_i/\partial\alpha_j$. Now $\partial\gamma_i/\partial\alpha_j = \gamma_i/\alpha_j$ when $j \succeq i$. When $j \prec i$, then from the proof of Theorem 2

$$\frac{\partial\gamma_i}{\partial\alpha_j} = A_i\frac{\partial\beta_i}{\partial\alpha_j} = A_i\frac{\beta_i}{\alpha_j}\prod_{m=1}^{\ell(j)-\ell(i)}\prod_{k\in d(f^m(j))\setminus f^{m-1}(j)}\overline{\beta}_k \tag{73}$$

From (68) (b), this goes to 0 as $\|\overline{\alpha}\| \to 0$. Finally, for all other j, γ_i does not depend on α_i, and so the derivative is 0. Summarizing, as $\|\overline{\alpha}\| \to 0$,

$$D(\alpha)^{-1} = \widetilde{D} + O(\|\overline{\alpha}\|) \quad \text{where} \quad \widetilde{D}_{ij} := \begin{cases} 1 & j \succeq i \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

Since matrix inversion is continuous in an open neighborhood of the non-singular matrices, then (72) follows if we can show that $\widetilde{D}_{ij}$ and D_{ij} are inverses. First $\sum_i D_{ki}\widetilde{D}_{ij} = \widetilde{D}_{kj} - \widetilde{D}_{f(k)j} = \delta_{kj}$ as required. Second $\sum_i \widetilde{D}_{ij}D_{jk} = \widetilde{D}_{ik} - \sum_{j\in d(k)}\widetilde{D}_{ik}$. The second term is only potentially non-zero when $k \succ i$. In this case the only term that contributes to the sum is when $j \succeq i$, giving -1. Hence $\sum_i \widetilde{D}_{ij}D_{jk} = \delta_{ik}$ as required.

(iv) By (iii), and continuity of finite dimensional matrix products, we have as $\|\overline{\alpha}\| \to 0$ that

$$\nu_{ik} = \sum_{j,j'} D_{ij}s(j \vee j')D_{kj'} + O(\|\overline{\alpha}\|^2). \tag{75}$$

It remains to evaluate

$$\sum_{j,j'} D_{ij} s(j \vee j') D_{kj'} = s(i \vee k) - s(i \vee f(k)) - s(f(i) \vee k) + s(f(i) \vee f(k)). \tag{76}$$

When $i = k$, then $i \vee k = i$, $i \vee f(k) = f(i) \vee f(k) = f(i)$ and so (76) yields $s(i) - s(f(i)) = \overline{\alpha}_i$, All other possible i and k yield zero, as we now show. If $i \prec k$ then $i \vee k = f(i) \vee k = k$, while $i \vee f(k) = f(i) \vee f(k) = f(k)$, and hence (76) is zero. The case $k \prec i$ is similar. In all other cases $i, k \prec i \vee k$ and so $i \vee k = i \vee f(k) = f(i) \vee k = f(i) \vee f(k)$. ∎

Proof of Theorem 8: Since $\alpha_k \in (0,1)$, each $Y_k(\cdot)$ is irreducible, and hence so is $Y(\cdot)$, and so q is the unique stationary distribution for Q, i.e. $\sum_z Q_{yz} q_z = q_y$. For n probes, $\widehat{\gamma}_j = \sum_{y \in \Phi(j)} \widehat{q}_y$ where $\widehat{q}_y = n^{-1} \sum_m \delta_{y\,Y(m)}$. By the Central Limit Theorem for Markov processes, see e.g. Chapter 17 of [15], $\widehat{q}$ is asymptotically Gaussian as $n \to \infty$ with

$$\sqrt{n}\,(\widehat{q} - q) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \xi) \tag{77}$$

where

$$\xi_{yz} = \lim_{n\to\infty} n\,\mathsf{Cov}(\widehat{q}_y, \widehat{q}_z) = \lim_{n\to\infty} n^{-1} \sum_{m=1}^{n} \sum_{m'=1}^{n} \mathsf{Cov}(\delta_{y\,Y(m)}, \delta_{z\,Y(m')}) \tag{78}$$

$$= q_y(\delta_{yz} - q_z) + 2 \sum_{m=1}^{\infty} (Q^m_{yz} - q_y) q_z . \tag{79}$$

∎

References

[1] J-C. Bolot and A. Vega Garcia "The case for FEC-based error control for packet audio in the Internet" ACM Multimedia Systems, to appear.

[2] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE '96*, October 1996.

[3] A. Dembo and O. Zeitouni, "Large deviations techniques and applications", Jones and Bartlett, Boston, 1993.

[4] B. Effron and D.V. Hinkley, "Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information", Biometrika, 65, 457–487, 1978.

[5] Felix: Independent Monitoring for Network Survivability. For more information see **ftp://ftp.bellcore.com/pub/mwg/felix/index.html**

[6] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, 1(4), August 1993.

[7] IPMA: Internet Performance Measurement and Analysis. For more information see **http://www.merit.edu/ipma**

[8] IP Performance Metrics Working Group. For more information see **http://www.ietf.org/html.charters/ippm-charter.html**

[9] V. Jacobson, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM '88, August 1988, pp. 314–329.

[10] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see **ftp://ftp.ee.lbl.gov/pathchar**

[11] E.L. Lehmann. "Theory of Point Estimation". Wiley-Interscience, 1983.

[12] B.N. Levine, S. Paul, J.J. Garcia-Luna-Aceves, “Organizing multicast receivers deterministically according to packet-loss correlation”, Preprint, University of California, Santa Cruz.

[13] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, “Creating a Scalable Architecture for Internet Measurement,” *to appear in Proc. INET ’98*.

[14] M. Mathis and J. Mahdavi, “Diagnosing Internet Congestion with a Transport Layer Performance Tool,” *Proc. INET ’96*, Montreal, June 1996.

[15] S.P. Meyn and R.L. Tweedie, “Markov Chains and Stochastic Stability”, Springer, New York, 1993.

[16] **mtrace** – Print multicast path from a source to a receiver. For more information see **ftp://ftp.parc.xerox.com/pub/net-research/ipmulti**

[17] **nam** – Network Animator. For more information see **http://www-mash.cs.berkeley.edu/ns/nam.html**

[18] **ns** – Network Simulator. For more information see **http://www-mash.cs.berkeley.edu/ns/ns.html**

[19] V. Paxson, “End-to-End Routing Behavior in the Internet,” *Proc. SIGCOMM ’96*, Stanford, Aug. 1996.

[20] V. Paxson, “Towards a Framework for Defining Internet Performance Metrics,” *Proc. INET ’96*, Montreal, 1996.

[21] V. Paxson, “End-to-End Internet Packet Dynamics,” *Proc. SIGCOMM 1997*, Cannes, France, 139–152, September 1997.

[22] V. Paxson, “Automated Packet Trace Analysis of TCP Implementations,” *Proc. SIGCOMM 1997*, Cannes, France, 167–179, September 1997.

[23] V. Paxson, “Measurements and Analysis of End-to-End Internet Dynamics,” Ph.D. Dissertation, University of California, Berkeley, April 1997.

[24] J. Postel, “Transmission Control Protocol,” RFC 793, September 1981.

[25] K. Ross & C. Wright, “Discrete Mathematics”, Prentice Hall, Englewood Cliffs, NJ, 1985.

[26] W. Rudin, “Functional Analysis”, McGraw-Hill, New York, 1973.

[27] M.J. Schervish, “Theory of Statistics”, Springer, New York, 1995.

[28] Surveyor. For more information see **http://io.advanced.org/surveyor/**

[29] K. Thompson, G.J. Miller and R. Wilder, “Wide-Area Internet Traffic Patterns and Characteristics,” *IEEE Network*, 11(6), November/December 1997.

[30] R.J. Vanderbei and J. Iannone, “An EM approach to OD matrix estimation,” Technical Report, Princeton University, 1994

[31] Y. Vardi, “Network Tomography: estimating source-destination traffic intensities from link data,” J. Am. Statist. Assoc., 91: 365–377, 1996.

[32] R.R. Wolff “Poisson Arrivals See Time Averages”, Operations Research, 30: 223–231, 1982

[33] M. Yajnik, J. Kurose, D. Towsley, “Packet Loss Correlation in the MBone Multicast Network,” *Proc. IEEE Global Internet*, Nov. 1996

Network Tomography on General Topologies *

Tian Bu
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
tbu@cs.umass.edu

Nick Duffield
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

Francesco Lo Presti
Dipartimento di Informatica
Università dell'Aquila
Via Vetoio, Coppito (AQ), Italy
lopresti@univaq.it

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@cs.umass.edu

ABSTRACT

In this paper we consider the problem of inferring link-level loss rates from end-to-end multicast measurements taken from a collection of trees. We give conditions under which loss rates are identifiable on a specified set of links. Two algorithms are presented to perform the link-level inferences for those links on which losses can be identified. One, the *minimum variance weighted average (MVWA) algorithm* treats the trees separately and then averages the results. The second, based on *expectation-maximization (EM)* merges all of the measurements into one computation. Simulations show that EM is slightly more accurate than MVWA, most likely due to its more efficient use of the measurements. We also describe extensions to the inference of link-level delay, inference from end-to-end unicast measurements, and inference when some measurements are missing.

1. INTRODUCTION

As the Internet grows in size and diversity, its internal behavior becomes ever more difficult to characterize. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data. Thus it is important to characterize internal performance from end-to-end measurements.

One promising technology that avoids these problems uses end-to-end multicast measurements from a single tree to infer link-level loss rates and delay statistics [1] by exploiting the inherent correlation in performance observed by multicast receivers. A shortcoming of this technology is that it is usually impossible to include all links of interest in any one tree. Consider the network in Figure 1(a) as an example. In this network, end-hosts 0 and 1 are sources, end-hosts 4 and 5 are receivers, and the set of links of interest is $\{(2,5) \quad (3,2)\}$. It is observed that both tree 1 and tree 2 are needed to cover the set of links of interest as illustrated in Figure 1(b) and 1(c). Therefore, in order to characterize the behavior of a network (or even a portion of it), it is necessary to perform measurements on multiple trees. Inferring link-level performance from measurements taken from several trees poses a challenging problem that is the focus of this paper.

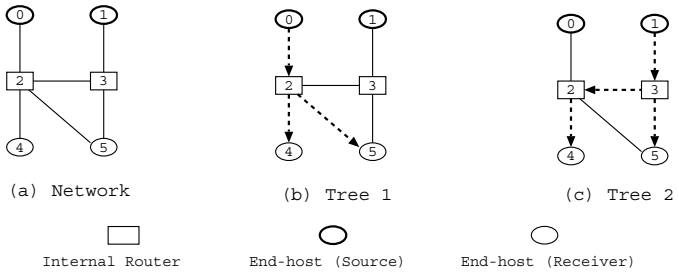


Figure 1: Single tree can not characterize a network

In this paper we address the following two problems. Given a collection of multicast trees, can we infer the performance of all of the links (or a specified subset) that are contained by the trees? Second, when the performance of the links of interest can be identified, how do we obtain accurate estimates of their performance? Focusing on loss rate as the performance metric, we introduce and evaluate two algorithms. The first, the *minimum variance weighted average (MVWA) algorithm*, performs inference on each tree separately and, for each link, returns a weighted average of the estimates taken from the different trees. This procedure may not always be able to infer the behavior of links whose loss rates are, nevertheless, identifiable. The loss rates for these links are obtained as a solution to a set of linear equations involving the inferred loss rates from individual trees. The second algorithm, the *expectation-maximization (EM) algorithm*, on the other hand, applies the standard expectation-maximization technique [15] to the measurement data taken from all of the trees. It returns estimates

*This work was supported in part by DARPA under contract F30602-00-2-0554 and F30602-98-2-0238, and by the National Science Foundation under Grant EIA-0080119.

of the loss rates of all identifiable links. We evaluate the two algorithms through simulation studying their convergence rates and relative performance. We find that EM estimates are at least as accurate than those produced by MVWA. The improvement is more pronounced when either the number or measurements is small or the distribution of measurements among the various trees is skewed.

Although the focus here is on link-level loss rates, we give extensions to EM to handle link delay. In addition, we show how MVWA and EM can be applied when end-to-end multicast measurements are not available, or when some measurements are missing.

There is a related problem of how to choose the set of trees so as to cover all of the links in the network (or subset of interest) in an efficient manner. This question has been dealt with elsewhere, [2] and is not considered here. We take as given the set of trees and observations from which we are to draw inferences.

Network tomography from end-to-end measurements has received considerable attention recently. In the context of multicast probing, the focus has been on loss, delay, and topology identification. Extensions to unicast probing can be found in [6, 7, 8, 11, 13]. However, these have treated only individual trees. There are techniques for round trip metrics such as loss rate and delay [14], based on measurements taken from a single node. Last, linear algebraic methods have been proposed for estimating link-level average round trip delays [19] and one-way delays, [12] . Neither of these extend to other metrics. Furthermore, the latter only yields biased estimates of average delays.

The remainder of the paper is organized as follows. Section 2 presents the model for a "multicast forest" (set of multicast trees). In Section 3 we present necessary and sufficient conditions for when the loss probabilities can be inferred from end-to-end multicast measurements. The MVWA and EM algorithms are presented in Section 4 along with convergence properties of the latter. Section 5 presents the results of simulation experiments. Extensions to delay inference, the use of unicast, and missing data are found in Section 6. Last, Section 8 concludes the paper.

2. NETWORK AND LOSS MODEL

Let $N = (V(N), E(N))$ denote a network with sets of nodes $V(N)$ and links $E(N)$. Here $(i,j) \in E(N)$ denotes a directed link from node i to node j in the network. Let Ψ denote a set of multicast trees embedded in N, i.e., $\forall T \in \Psi$, $V(T) \subseteq V(N)$ and $E(T) \subseteq E(N)$. We denote $\cup_{T\in\Psi} V(T)$ by $V(\Psi)$ and $\cup_{T\in\Psi} E(T)$ by $E(\Psi)$. Note that $(i,j) \in E(\Psi)$ can appear in more than one tree. For $(i,j) \in E(N)$, we denote $\Psi_{i,j} \subseteq \Psi$ the set of trees which include link (i,j). Consider a tree $T \in \Psi$. Each node i in T, apart from the root $\rho(T)$, has a parent in T, $f(i,T)$, such that $(f(i,T),i) \in E(T)$. The set of children of i in tree T is denoted by $d(i,T)$. Let $\tau_{i,T}$ denote the subtree of T rooted at node i. Let $R(\tau_{i,T})$ denote the receivers in subtree $\tau_{i,T}$. We denote the path from node i to j, $i,j \in V(T)$ in tree T by $p_T(i,j)$. Define a segment in T to be a path between either the root and the closest branch point, two neighboring branch points, or a branch point and a leaf. We represent a segment by the set of links that comprises it.

For $T \in \Psi$, we identify the root $\rho(T)$ with the source of probes, and the set of leaves $R(T)$ with the set of receivers. For a tree T, a probe is sent down the tree starting at the root. If it reaches node $j \in V(T)$, a copy of the probe is produced and sent down the tree toward each child of j. As a packet traverses link (i,j), it is lost with probability $1-\alpha_{i,j}$ and arrives at j with probability $\alpha_{i,j}$. We denote $1-\alpha_{i,j}$ by $\overline{\alpha}_{i,j}$. Let $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. We assume losses of the same probe on different links and of different probes on the same link are independent, and that losses of probes sent from the different sources $\rho(T)$, $T \in \Psi$ are independent.

We describe the passage of probes down each tree T by a stochastic process $X_T = (X_{k,T})_{k\in V(T)}$ where $X_{k,T} = 1$ if the probe reaches node k, 0 if does not. By definition $X_{\rho(T),T} = 1$. If $X_{i,T} = 0$ then $X_{j,T} = 0$ for all $j \in d(i,T)$. If $X_{i,T} = 1$ then for $j \in d(i,T)$, $X_{j,T} = 1$ with probability $\alpha_{i,j}$ and $X_{j,T} = 0$ with probability $\overline{\alpha}_{i,j}$. We assume that the collection of trees is in *canonical form*, namely that $0 < \alpha_{i,j} < 1, \forall (i,j) \in E(\Psi)$. An arbitrary collection of trees can be transformed into one with canonical form.

In an experiment, a set of probes is sent from the multicast tree sources $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we can think of each probe as a trial, the outcome of which is a record of whether or not the probe was received at each receiver in $R(T)$. In terms of the random process X_T, the outcome is a configuration $X_{R(T)} = (X_{i,T})_{i\in R(T)}$ of zeros and ones at the receivers. Notice that only the values of X_T at the receivers are observable; the values at the internal nodes are unknown. Each outcome is thus an element of the space $\Omega_{R(T)} = \{0,1\}^{\#R(T)}$. For a given set of link probabilities α the distribution of $X_{R(T)}$ on $\Omega_{R(T)}$ will be denoted $\mathsf{P}_{\alpha,T}$. The probability of a single outcome $x \in \Omega_{R(T)}$ is $p(x;\alpha) = \mathsf{P}_{\alpha,T}[X_{R(T)} = x]$.

3. IDENTIFIABILITY

In order to perform tomography from measurements on the tree set Ψ, we require that the link probabilities are determined from the set leaf probabilities that are measured directly. We phrase this in terms *identifiability*, which captures the property that link probabilities can be distinguished by measurements from an infinite sequence of probes. We say that $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α if for any α', $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi} = \{\mathsf{P}_{\alpha',T}\}_{T\in\Psi}$ implies $\alpha = \alpha'$. In this section, we establish necessary and sufficient conditions for identifiability.

We are given a set of canonical trees Ψ with an associated link success probability vector $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. Let S be the set of all segments within the trees contained in Ψ. Define β_s to be the logarithm of the probability that a packet successfully traverses segment $s \in S$ given that it reached the start of that segment, $\beta_s = \log(\prod_{(i,j)\in s} \alpha_{i,j}) = \sum_{(i,j)\in s} \log \alpha_{i,j}$. We introduce the $\#S \times \#E(\Psi)$ matrix A where $A_{s,(i,j)} = 1$ if link (i,j) belongs to segment s and 0 otherwise. Using the sets of trees in Figure 1 as an example, if we order the links as $(0,2)$ $(2,4)$ $(2,5)$ $(1,3)$ $(3,5)$ $(3,2)$ and the segment as $\{(0,2)\}$ $\{(2,4)\}$ $\{(2,5)\}$ $\{(1,3)\}$ $\{(3,5)\}$ $\{(3,2),(2,4)\}$, the matrix A is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

If we define $z_{(i,j)} = \log \alpha_{i,j}, \forall (i,j) \in E(\Psi)$, we then have the following equation

$$Az = \beta \tag{1}$$

Here the components of z are $z_{(i,j)}$ and the components of β are β_s. Note that A needs not be a square matrix in general.

Before stating and proving results on identifiability, we note that for a given set of link probabilities α, there exists at least one solution, namely $z = \log \alpha$, to (1). Let A^T denote the matrix transpose of A.

THEOREM 1. *Let Ψ be a set of canonical loss trees. Then the following are equivalent:*

(i) For some α, $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α.

(ii) Equation (1) has a unique solution $z = (A^T A)^{-1} A^T \beta$.

(iii) $Az = 0$ iff $z = 0$.

(iv) For all α, $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies α.

Proof. (i)⇔(ii). First, we note that β is identifiable from $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ (Theorem 3 in [4]). Suppose that $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ cannot identify α, i.e., there are at least two sets of link probabilities, α and α' that are consistent with $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. Based on the derivation of (1) there cannot exist an unique solution to (1). Similarly, if α is identifiable, it is obtained by solving (1). Suppose that (1) does not have a unique solution. Then, from the derivation of (1) it follows that there exist multiple values of α that can give rise to $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. Suppose that there exists a unique solution to (1). It is easy to show by contradiction that necessarily there is only one value of α that can give rise to $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$. For (ii) ⇔(iii), observe that (1) has a unique solution if and only if the nullspace of A is in $\{0\}$. In this case $A^T A$ is invertible, and the expression for z then follows on pre-multiplying (1) by A^T. $(A^T A)^{-1} A^T$ is the generalized inverse of A; see [16]. Furthermore, solutions of (1) must be unique for all α, and hence (ii)⇔(iv). □

It should be clear from this theorem that identifiability is a topological property, i.e., not dependent on the values α. We can use this fact to select β at our convenience. Suppose we are interested in identifying a set of links a set of links $C \subset E(\Psi)$. Choosing $\alpha_{i,j} = e^{-1}, \forall (i,j) \in E(\Psi)$ results in $\beta_s = \#s$. Hence we have:

THEOREM 2. *Let Ψ be a set of canonical loss trees. $\{\mathsf{P}_{\alpha,T}\}_{T\in\Psi}$ identifies $(\alpha_{i,j})_{(i,j)\in C}$ iff there is a unique value of $\{z_{(i,j)} : (i,j) \in C\}$ that satisfies equation (1) for $\beta_s = \#s, \forall s \in S$.*

4. LOSS INFERENCE

In this section, we describe two algorithms for loss inference in a collection of multicast trees. In the first algorithm we perform inference on each tree separately, and then we take the weighted average of the different estimates so obtained. In the second algorithm we perform inference on the entire set of measurement from all of the trees using the Expectation-Maximization (EM) algorithm.

4.1 Measurement Experiment

A measurement experiment for a collection of multicast trees Ψ consists of sending n_T probes from $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we denote by $\mathbf{x}_{R(T)} = (x^1_{R(T)}, \ldots, x^{n_T}_{R(T)})$, (with $x^m_{R(T)} = (x^m_{k,T})_{k\in R(T)}$) the set measured of end-to-end loss down T. $\mathbf{x}_R = (\mathbf{x}_{R(T)})_{T\in\Psi}$ will denote the complete set of measurements.

4.2 Minimum Variance Weighted Average

A technique for loss inference for a single tree has been proposed in [4]. For a given set of trees Ψ, we can proceed as follows: (1) consider each tree $T \in \Psi$ separately, by using the algorithm provided in [4] on the measurements $\mathbf{x}_{R(T)}$; this yields estimates for all segments in T; (2) combine the estimates from the different trees.

We first consider the problem of combining estimators of segment transmission probabilities. Let s be a segment, and $\Psi_s \in \Psi$ the maximal set of topologies that include s as segment. Inference on each logical topology $T \in \Psi_s$ provides us with an estimate $\hat{q}_{s,T}$ of the transmission probability $q_s = e^{\beta_s}$ across the segment s. How should the $\hat{q}_{s,T}$ be combined to form a single estimate of q_s?

We consider convex combinations of the form

$$\hat{q}_s = \sum_{T\in\Psi_s} \lambda_T \hat{q}_{s,T}, \quad \lambda_T \in [0,1]; \sum_{T\in\Psi_s)} \lambda_T = 1. \tag{2}$$

We propose to select the minimum variance combination as the single estimator. By assumption, the $\hat{q}_{s,T}$ are independent, and so

$$\mathsf{Var}(\hat{q}_s) = \sum_{T\in\Psi_s} \lambda_T^2 \mathsf{Var}(\hat{q}_{s,T}). \tag{3}$$

$\mathsf{Var}(\hat{q}_{s,T})$ is clearly jointly convex in the $(\lambda_T)_{T\in\Psi_s}$, and by explicit differentiation under the constraint $\sum_{T\in\Psi_s} \lambda_T = 1$, the minimum for $\mathsf{Var}(\hat{q}_s)$ occurs when

$$\lambda_T = \frac{\mathsf{Var}(\hat{q}_{s,T})^{-1}}{\sum_{T'\in\Psi_s} \mathsf{Var}(\hat{q}_{s,T'})^{-1}} \tag{4}$$

Now, in general, $\mathsf{Var}(\hat{q}_{s,T})$ depends on the topology T. But it follows from Theorem 5 in [4] that the asymptotic variance $n_T \mathsf{Var}(\hat{q}_{s,T})$ converges to $\overline{q}_s + O(\|\overline{\alpha}\|^2)$ as $n_T \to \infty$. Thus, for small loss probabilities, we can use the approximation $\mathsf{Var}(\hat{q}_{s,T}) \approx n_T^{-1} \overline{q}_s$. In this approximation, the coefficients $\lambda_T \approx n_T / \sum_{T'\in\Psi(T)} n_{T'}$. We will use this approximation in (2) as our minimum variance weighted average algorithm (MVWA) algorithm, i.e.,

$$\hat{q}_s = \frac{\sum_{T\in\Psi_s} n_T \hat{q}_{s,T}}{\sum_{T\in\Psi_s} n_T} \tag{5}$$

We note two special cases: (i) s comprises a single link (i,j), in which case the estimate is for the link rate $\alpha_{i,j}$; (ii) only one tree contains s, in which case the sums in (5) trivially have one term.

It remains to recover link probabilities from the $\hat{q}_s$. Following Theorem 1, identifiable link probabilities $\alpha_{i,j}$ are estimated by

$$\log \hat{\alpha}_{i,j} = \sum_s A^*_{(i,j),s} \log \hat{q}_s \tag{6}$$

A simple example is when two segments s, s' are such that s is obtained by appending the link (i,j) to s'. Clearly $A^*_{(i,j),s} = 1 - A^*_{(i,j),s'}$ with (6) reducing to taking quotients: $\hat{\alpha}_{i,j} = \hat{q}_s / \hat{q}_{s'}$.

4.3 EM Algorithm

Here we turn to a more direct approach to inference, namely, we use the Maximum Likelihood Estimator to estimate α from the set of measurements $\mathbf{x}_R$, i.e., we estimate α by the value $\widehat{\alpha}$ which maximizes the probability of observing $\mathbf{x}_R$.

Let $n_T(x_{R(T)})$ denote the number of probes for which the outcome $x_{R(T)} \in \Omega_{R(T)}$ is obtained, $T \in \Psi$. The probability of the n_T

independent observations $\mathbf{x}_{R(T)}$ is then

$$p(\mathbf{x}_{R(T)};\alpha) = \prod_{m=1}^{n_T} p(x^m_{R(T)};\alpha) = \prod_{x_{R(T)}\in\Omega_{R(T)}} p(x_{R(T)};\alpha)^{n_T(x_{R(T)})}$$

and the probability of the complete set of measurement $\mathbf{x}_R$ at the receivers is

$$p(\mathbf{x}_R;\alpha) = \prod_{T\in\Psi} p(\mathbf{x}_{R(T)};\alpha). \quad (7)$$

Our goal is to estimate α by the maximizer of (7), namely,

$$\widehat{\alpha} = \arg\max p(\mathbf{x}_R;\alpha). \quad (8)$$

In [4], a direct expression for $\widehat{\alpha}$ are obtained for the case of a single tree, *i.e.*, when $\#\Psi = 1$. For the general case, unfortunately, we have been unable to obtain a direct expression for $\widehat{\alpha}$. Instead, we follow the approach in [7, 8], and employ the EM algorithm to obtain an iterative approximation $\widehat{\alpha}^{(\ell)}$, $\ell = 0, 1, \ldots,$ to $\widehat{\alpha}$. To understand the idea behind the EM algorithm, assume that we can observe the entire loss process at each node, i.e., assume knowledge of the values $\mathbf{x}_T = (x^1_T, \ldots, x^m_T)$, (with each $x^m_T = (x^m_{k,T})_{k\in V(T)}$), $T \in \Psi$. In this case estimation of α becomes trivial: with complete data knowledge it is easy to realize that the MLE estimate of the success probability $\alpha_{i,j}$ along link (i,j), $\widehat{\alpha}_{i,j}$, is just the fraction of probes successfully transmitted along (i,j), $(i,j) \in E(\Psi)$, i.e.,

$$\widehat{\alpha}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} n_{j,T}}{\sum_{T\in\Psi_{i,j}} n_{i,T}} \quad (i,j) \in E(\Psi), \quad (9)$$

where $n_{k,T} = \sum_{m=1}^{n_T} x^m_{k,T}$ is the number of probes sent from $\rho(T)$ which arrived to node $k \in V(T)$, $T \in \Psi$.

The EM algorithm assumes complete knowledge of the loss process such that the resulting likelihood has a simple form. Since the complete data, and thus the counts $n_{k,T}$ (except for the leaves nodes) are not known, the EM algorithm proceeds iteratively to augment the actual observations with the unobserved observation at the interior links. Below we briefly describe the algorithm and the intuition behind it. We spell out the detail in Section7.

- *Step 1.* Select an initial link loss rate $\widehat{\alpha}^{(0)}$. The simulation study suggests the values that the algorithm converges to are independent of $\widehat{\alpha}^{(0)}$.
- *Step 2.* Estimate the (unknown) counts $n_{k,T}$ by $\hat{n}_{k,T} = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{k,T}|\mathbf{x}_R]$. In other words, we estimate the counts by their conditional expectation given the observed data $\mathbf{x}_R$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$.
- *Step 3.* Compute the new estimate $\alpha^{(\ell+1)}$ via (9), using the estimated counts $\hat{n}_{k,T}$ computed in the previous step in place of the actual (unknown) counts $n_{k,T}$. In other words, we set

$$\widehat{\alpha}^{(\ell+1)}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} \hat{n}_{j,T}}{\sum_{T\in\Psi_{i,j}} \hat{n}_{i,T}} \quad (i,j) \in E(\Psi). \quad (10)$$

- *Step 4.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Tree	Source	Receivers
1	0	12 13 14 15 16 17 18 19
2	1	12 13 14 15 16 17 18 19
3	2	12 13 14 15
4	25	16 17 18 19

Table 1: Tree layout for model simulation

As shown in Section7, the EM iterates converges to a local (but not necessarily) global maximizer of (7). However, our simulation results suggests it always converge to the global maximizer $\widehat{\alpha}$ and the convergence does not depend on the initial values.

5. SIMULATION EVALUATION

We evaluate our loss inference algorithms using the `ns` [18] simulator. This work has two parts: model simulation and network simulation. In the model simulation, losses are determined by time-invariant Bernoulli processes. In the network simulation, losses are due to congestion as probes compete with other background traffic. The majority of the background traffic in the network simulation is produced by TCP flows. However, we do include some on-off flows where the on and off periods have either a Pareto or an exponential distribution. We chose such a mix because TCP is the dominant transport protocol on the Internet.

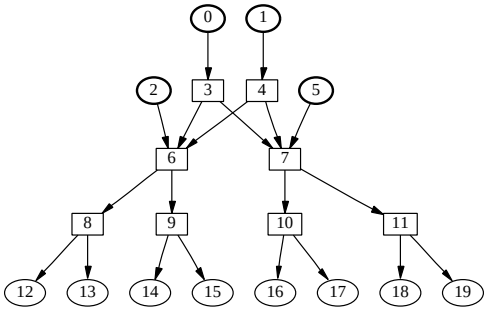


Figure 2: Model simulation topology: Nodes are of three types; bold ellipse: potential sender, ellipse: potential receivers, and box: internal nodes.

5.1 Comparing loss probability

Our approach for comparing two sets of loss probabilities was first introduced in [5]. Assume that we want to compare two loss probabilities p and q. For example p could be an inferred probability on a link, q the corresponding actual probability. For some **error margin** $\varepsilon > 0$ we define the **error factor**

$$F_\varepsilon(p,q) = \max\left\{\frac{p(\varepsilon)}{q(\varepsilon)}, \frac{q(\varepsilon)}{p(\varepsilon)}\right\} \quad (11)$$

where $p(\varepsilon) = \max\{\varepsilon, p\}$ and $q(\varepsilon) = \max\{\varepsilon, q\}$. Thus, we treat p and q as being not less than ε, and having done this, the error factor is the maximum ratio, upwards or downwards, by which they differ. Unless otherwise stated, we used the default value $\varepsilon = 10^{-3}$ in this paper. This choice of metric is motivated by the desire to estimate the relative magnitude of loss ratios on different links in order to distinguish those which suffer higher loss.

5.2 Model simulation

The topology for model simulation is presented in Figure 2. A total of four trees are embedded in the topology as described in Table 1. A time-invariant Bernoulli loss processes is associated with each link. In the simulation, uniform loss rates are assigned to all links.

We use loss rates of 2% and 4% on each link and let each source send equal numbers of probes down to the trees. For each loss rate, we vary the total number of probes sent by all sources from 50 to 1600. Each setting is simulated ten times with different random seeds. For each simulation, we use both the MVWA and EM to estimate loss rates and compare with the actual simulation loss rates.

Figure 3 shows box-plots[1] of error factors between inferred loss and simulated loss over all links and all runs. In the figure, error factors are displayed as a function of number of probes and one graph is for each loss rate. (Note that the total number of probes increase exponentially). In each graph, we plot error factors for both MVWA (abbreviated as WA) algorithm and EM algorithm. Observed from graph that the estimates produced by EM algorithm show greater accuracy and less variability than these produced by MVWA algorithm under both loss rates we simulate when the number of probes are small. However, as the number of probes increases, the estimates yielded by both algorithm become more accurate, the difference between two algorithm become less, and their variability reduces. The same set of simulations were done when the numbers of probes in each tree are different. The results are very close to the case where the numbers of probes are equal.

Note that every link in the topology described in Figure 2 is a segment in at least one of the trees. We also simulated a network embedded by a collection of trees where some links are not a segment in any trees even they are identifiable. The error factors we observed are very similar to those presented in Figure 3.

Since the EM algorithm is more accurate and of less variability than MVWA algorithm, we focus on evaluating EM algorithm in next subsection.

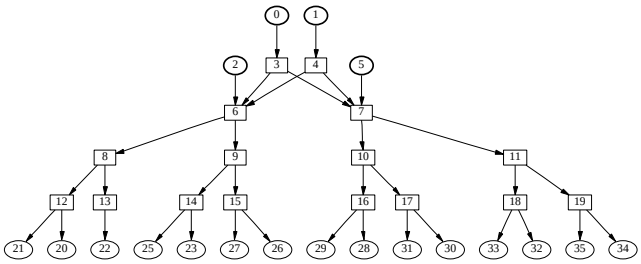


Figure 4: Small network simulation topology: Nodes are of three types; bold ellipse: potential sender, ellipse: potential receivers, and box: internal nodes.

5.3 Network simulation

In this section, we simulate two topologies, a small network in Figure 4 and a multicast topology based on the Abilene network. In both topologies, background traffic is generated by infinite TCP and on-off UDP flows. All the routers in the network are configured to be droptail routers since the droptail routers are prevalent in the Internet.

[1] In a box-plot, the box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers.

Tree	Source	Receivers
1	0	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
2	1	20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
3	2	20 21 22 23 24 25 26 27
4	5	28 29 30 31 32 33 34 35

Table 2: Tree layout for small network simulation

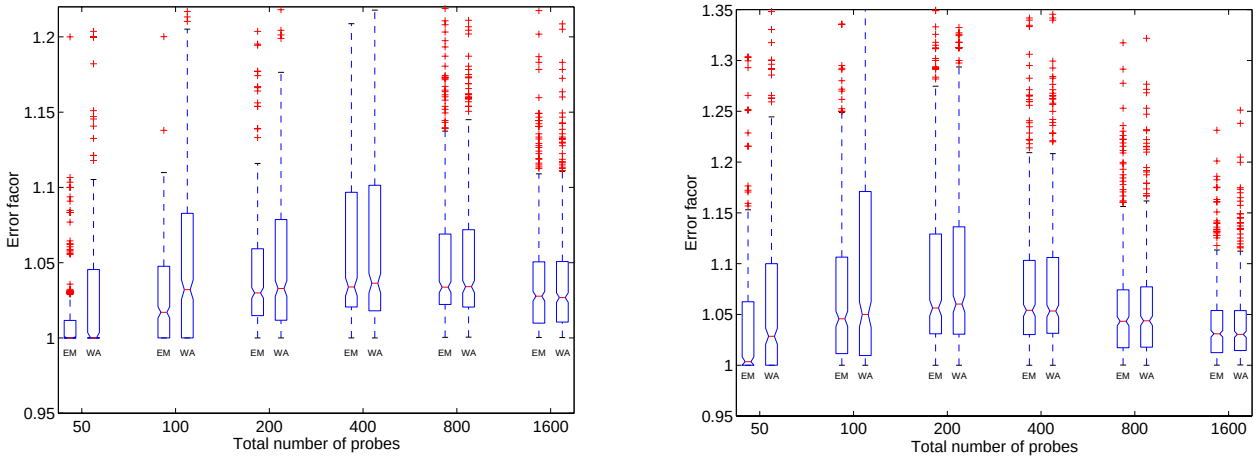
Small network. The tree layout of the small network is described in Table 2. We use constant bit rate probes and the interval between probes is $100ms$. We conducted a total of 7 simulations which differ according to the duration of the measurement. We start with an initial duration of 2 seconds and double it each time until reaching 128 seconds. Each of these simulations is run 10 times with different random seeds. For each simulation, we calculate the loss rates using the EM algorithm.

The link losses in the set of simulations are due to all flows competing for bandwidth. Since different types of flows may exhibit different behavior, the probe flow does not necessarily suffer the same loss rate as the background flows do. Therefore, the error of using inferred loss to estimate the link loss may due to one of the two possibilities. Either probe traffic loss rate differ from all traffic loss rate or the estimates yielded by the EM algorithm do not agree with the probe loss rate. In order to distinguish them, we compare the inferred results to both probe loss rate and all traffic loss rate.

Figure 5 illustrates box plots of error factors for all links and all simulation runs. The error factors are plotted as a function of measurement time. On the left we show the error factor between inferred and simulated all traffic loss; on the right between inferred and simulated probe loss. We observe from both graphs in the figure that both the error factors and their variabilities decrease as the number of probes increase. The improvements are more significant for short measurements.

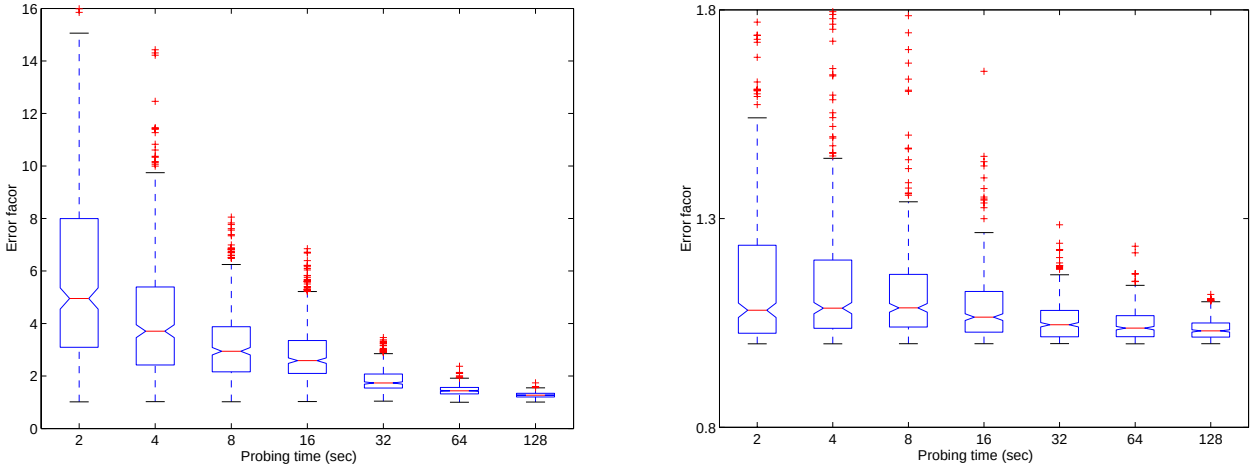
We present scatter plots for the all traffic loss vs. inferred loss on the left and probe traffic loss vs. inferred loss on the right in Figure 6 when the measurement duration is 128 seconds. We make two observations. First, the inferred loss rate almost always overestimates the link loss rate. Second, the inferred loss rate provides a very good estimate of the probe traffic loss rate. The difference between the inferred loss rates and all traffic loss rates is due to that the probe traffic endures a higher loss rate than the rest of traffic. We conjecture that this is because the majority of the background traffic come from infinite TCP flows. TCP reduces its sending rate when the losses are detected. Therefore, fewer TCP packets will suffer loss. However, the CBR source sends probes at a constant rate which is not affected by congestion. We expect the algorithm to be more accurate in the Internet since the Internet contains many short lived TCP flows and many of them complete transmission before they respond to losses.

Abilene network. Abilene [21] is an advanced backbone network that supports the work of Internet2 universities as they develop advanced Internet applications. One major goal of Abilene is to provide a separate network to enable the testing of advanced network capabilities prior to their introduction into the application develop-



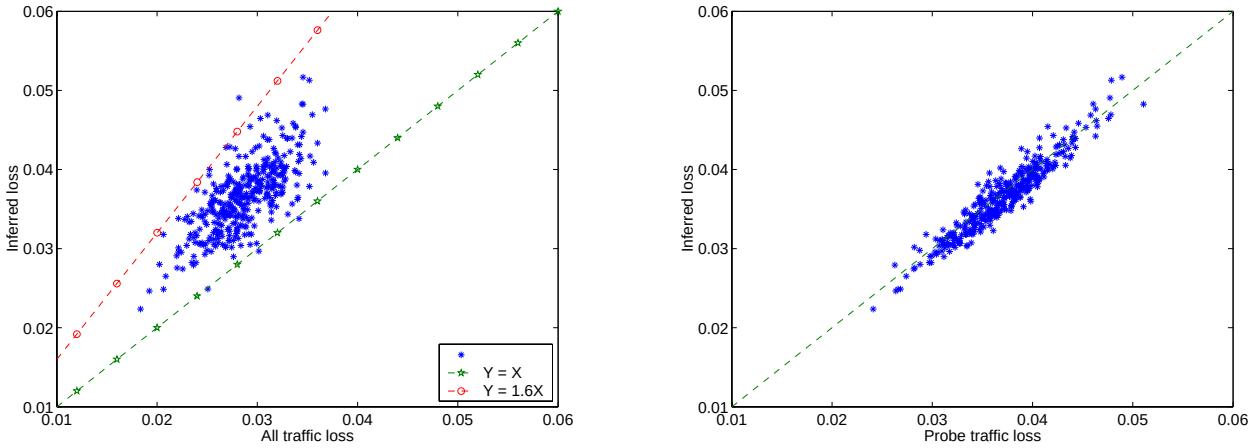
(a) loss = 2% (b) loss = 4%

Figure 3: Accuracy of MVWA(WA) algorithm vs. EM: Box-plot of error factors over all links and all runs for loss rate 2%(left) and 4%(right).



(a) All loss vs. inferred loss (b) Probe loss vs. inferred loss

Figure 5: Accuracy of EM algorithm vs. probing time: Error factor over all links and all runs



(a) All loss vs. inferred loss (b) Probe loss vs. inferred loss

Figure 6: Small network scatter plot: inferred loss vs. all loss, inferred loss vs. probe loss

ment network. Multicast is one among all such services. Abilene supports native and sparse mode multicast. As of October 01, 2001, Multicast protocols, PIM-sparse, MBGP and MSDP have been deployed in the backbone. The Abilene multicast logical topology is illustrated in [22]. It consists of 159 nodes and 165 edges. Each node in the graph represents a physical location and each link represents a physical interconnection between some two routers from different locations. Because the more detailed topology within each physical location is not available to us, we treat each node as a router and focus on the logical topology in our experiments. There are three types of links in Abilene backbone, OC3 (155M), OC12 (622M) and OC48 (2.5G). The type of the links that connect participants to backbone are not labeled and we assume they are T3 (45M). Since the `ns` simulator does not allow us to simulate enough number of flows to fill up such high bandwidth links and generate losses, we scale down the bandwidth proportionally by 10^8 times. Last, we assume that only the leaves in the topology (i.e., node of degree one) are senders or receivers.

We lay out a total of eight trees that can identify 41 links. An equal number of probes is sent by each source and the interval between probes is $200ms$. We conducted a simulation of duration 256 seconds and ran it ten times with different random seeds. For each simulation, we estimate the loss rates using the EM algorithm and compare them to the simulated loss rates. Figure 7 illustrates scatter plots for inferred loss vs. all loss (left) and inferred loss vs. probe loss (right). Similar to what we observed in small network simulation, the EM algorithm provides accurate estimates of probe loss rates. However, the inferred loss rates are almost always higher than the simulated all traffic loss rates.

6. EXTENSIONS

In this section, we first extend the EM algorithm to infer the distribution of links delay. Second, since multicast is not supported everywhere in the Internet and internal performance observed by multicast packets may differ from that observed by unicast packets, it is important to show our algorithms for inferring a set of multicast trees can be applied to unicast measurements. Last, the algorithms we presented so far rely on the availability of complete information from the receivers. However, this may pose a serious problem in their deployment. We demonstrate the use of our algorithms to handle incomplete observations from receivers.

6.1 Delay inference

We now illustrate the use of end-to-end measurements from a collection of multicast trees Ψ to estimate the delay characteristics of internal links.

We associate with each link (i,j) a random variable $D_{i,j}$ which represents the queueing delay that would be encountered by packets traversing link (i,j). For the analysis, we quantize the queueing delay to a finite set of values $\mathcal{Q} = \{0, q, 2q, \ldots, Bq, \infty\}$, where q is a suitable fixed bin size. A queueing delay equal to ∞ indicates that the packet is lost on the link. We define the bin associated to $iq \in \mathcal{Q}$ to be the interval $[iq - q/2, iq + q/2), i = 1, \ldots, B$, and $[Bq + q/2, \infty)$ the one associated to the value ∞. Because delay is non-negative, we associate with 0 the bin $[0, q/2)$. We thus model the link queueing delay by a nonparametric discrete distribution that we can regard as a discretized version of the actual delay distribution. We denote the distribution of $D_{i,j}$ by $\alpha_{i,j} = (\alpha_{i,j}(d))_{d\in\mathcal{Q}}$, where $\alpha_{i,j}(d) = P[D_{i,j} = d], d \in \mathcal{Q}$. We will denote $\alpha = (\alpha_{i,j})_{(i,j)\in E(\Psi)}$. We assume that queueing delays are independent between different packets, and for the same packets on different links. Thus the progress of each probe down the tree T is described by an independent copy of a stochastic process $Y_T = (Y_{k,T})_{k\in V(T)}$ which represents the accrued queueing delay of packets. The queueing delay experienced by a packet from $\rho(T)$ to node i is $Y_{i,T} = \sum_{(m,n)\in p_T(\rho(T),i)} D_{m,n}$ where $p_T(\rho(T), i)$ denote the path on tree T from source to node i.

In an experiment, a set of probes is sent from the multicast tree sources $\rho(T)$, $T \in \Psi$. For each $T \in \Psi$, we can think of each probe as a trial, the outcome of which is a configuration of source to receivers queueing delays $Y_{R(T)} = (Y_{k,T})_{k\in R(T)}$ we also discretize to the set $\mathcal{Q}$. Each outcome is thus an element of the space $\Omega_{R(T)} = \mathcal{Q}^{\#R(T)}$.

As with loss estimation, we use maximum likelihood estimation based on measurements across the multicast tress $T \in \Psi$. Let us dispatch n_T probes from $\rho(T)$, $T \in \Psi$, and let $n_T(y_{R(T)})$ denote the number of probes for which the outcome $y_{R(T)} \in \Omega_{R(T)}$ is obtained. The probability of the n_T independent observations $\mathbf{y}_{R(T)} = (y^1_{R(T)}, \ldots, y^{n_T}_{R(T)})$, (with each $y^m_{R(T)} = (y^m_{k,T})_{k\in R(T)}$), is then

$$\begin{aligned} p(\mathbf{y}_{R(T)};\alpha) &= \prod_{m=1}^{n_T} p(y^m_{R(T)};\alpha) \\ &= \prod_{y_{R(T)}\in\Omega_{R(T)}} p(y_{R(T)};\alpha)^{n_T(y_{R(T)})} \end{aligned}$$

where $p(y;\alpha) = \mathsf{P}_\alpha[Y_T = y_T]$. The probability of the complete set of measurements $\mathbf{y}_R = (\mathbf{y}_{R(T)})_{T\in\Psi}$ at the receivers is

$$p(\mathbf{y}_R;\alpha) = \prod_{T\in\Psi} p(\mathbf{y}_{R(T)};\alpha). \tag{12}$$

Our goal is to estimate α by the maximizer of (12), namely,

$$\widehat{\alpha} = \arg\max p(\mathbf{y}_R;\alpha). \tag{13}$$

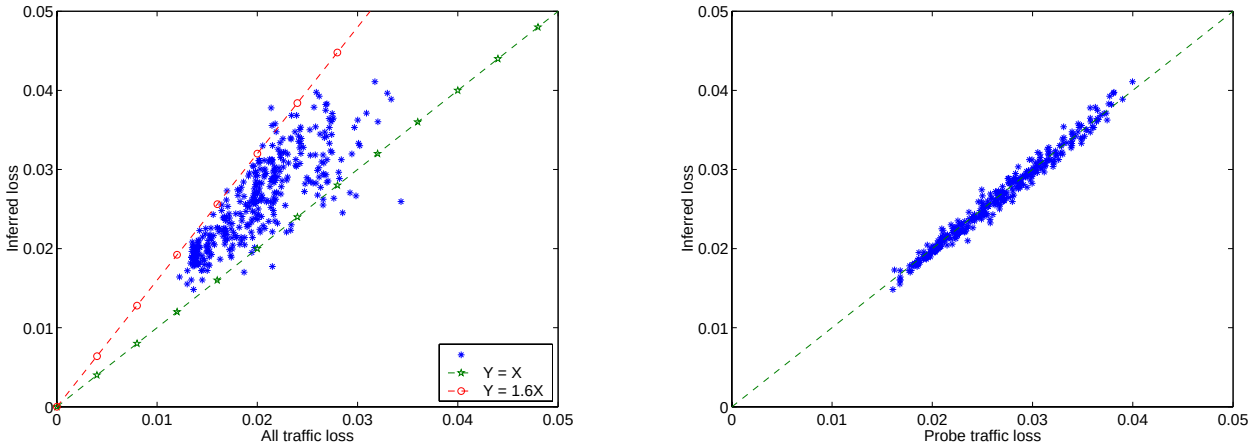
As with loss inference, we resort to the EM algorithm to obtain an iterative solution $\widehat{\alpha}^{(\ell)}$, $\ell = 0, 1, \ldots$, to a (local) maximizer of the likelihood (12). Assume complete knowledge of the delay process at each link, namely the values $\mathbf{y}_T = (y^1_T, \ldots, y^m_T)$, (with each $y^m_T = (y^m_{k,T})_{k\in V(T)}$), $T \in \Psi$. Denote by $n_{i,j,T}(d)$ the total number of packets sent by $\rho(T)$ that experienced a delay equal to d along link (i,j). It is easy to verify that with complete data, the MLE estimate of $\alpha_{i,j}(d)$ is

$$\widehat{\alpha}_{i,j}(d) = \frac{\sum_{T\in\Psi_{i,j}} n_{i,j,T}(d)}{\sum_{T\in\Psi_{i,j}}\sum_{d\in\mathcal{Q}} n_{i,j,T}(d)} \quad \forall (i,j) \in E(\Psi). \tag{14}$$

Thus, with complete knowledge, the MLE estimate of $\alpha_{i,j}(d)$ is simply the fraction of the probes traversing link (i,j) which encountered a delay equal to d.

For delay inference the EM algorithm proceeds as for the loss case. Below we briefly describe the algorithm and intuition behind it. Details can be found in [3].

1. *Step 1.* Select the initial link delay distribution $\widehat{\alpha}^{(0)}$.
2. *Step 2.* Given the current estimate $\widehat{\alpha}^{(\ell)}$, Estimate the (unknown) counts $n_{i,j,T}(d)$ by $\hat{n}_{i,j,T}(d) = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{i,j,T}(d)|\mathbf{y}_R]$. In other words, we estimate the counts by their conditional expectation given the observed data $\mathbf{y}_R$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$.



(a) All loss vs. inferred loss

(b) Probe loss vs. inferred loss

Figure 7: Abilene scatter plot: inferred loss vs. all loss, inferred loss vs. probe loss

3. *Step 3.* Compute the new estimate $\alpha^{(\ell+1)}$ via (14), using the estimated counts $\hat{n}_{i,j,T}(d)$ computed in the previous step in place of the actual (unknown) counts $n_{i,j,T}(d)$.

4. *Iteration.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Complexity
The complexity of the algorithm is dominated by the computation the conditional expectations which can be accomplished in time linear with $\#V(T) \times \#\mathcal{Q}$, $T \in \Psi$. The computation can be done by extending the approach for computing loss conditional probability and is described in [3].

Convergence
The conditions for convergence can be established similarly as for loss inference.

6.2 Inference with unicast measurement

So far we have presented inference algorithms for a collection of trees based on end-to-end multicast measurements. These techniques can be extended to work with unicast measurements from multiple sources as well.

The rationale behind unicast based inference is that: (1) measurement domain is limited because large portions of the Internet do not support network-level multicast, and that (2) the internal performance observed by multicast packets may differs from that observed by unicast packets. Techniques for unicast measurements and inference have been recently proposed in [6, 11] for the inference of loss rates and [7, 8, 9] for delay distributions. However, these works only handle the inference of a *single* source with multiple pairs of receivers and thus may pose severe limitations in scope.

The key idea behind unicast inference is to design unicast measurement whose correlation properties closely resemble those of multicast traffic, so that it becomes possible to use the inference techniques developed for multicast inference; the closer the correlation properties are to that of multicast traffic, the more accurate the results.

A basic approach for unicast inference is to dispatch two back-to-back packets (a packet pair) from a probe source to a pair of distinct receivers. For each such packet pair, the two packets traverse a common set of links down a node where their paths diverge to the two receivers. By choosing *multiple* sources and pairs of receivers, it is possible to cover a more significant portion of a network than with a single source. The inference for the link loss probability and link delay distribution from a set of packet pair measurements is formulated as a maximum likelihood estimation problem which is then solved using the algorithms we presented earlier in the paper. The idea, is that treat the unicast packet pair measurements as statistically equivalent to a notion multicast packet that descends the same tree. The entire set of measurements is thus considered equivalent to a set of multicast measurements down a collection of 2 leaf trees. The analysis then follows the same approach for a collection of trees detailed in Sections 4 and 6.1.

6.3 Inference with missing data

The algorithms presented in the paper so far rely on the availability of complete information from the receivers. However, as described in [10], this may pose a serious problem in their deployment. For example, the loss reports from receives may be delivered unreliably and there may be bandwidth constraints for transmitting loss reports. Therefore, it is important to extend the algorithms to handle incomplete data sets. An algorithm has been proposed in [10] to handle incomplete data for a single tree. The goal of this section is to extend the algorithms we proposed earlier in the paper to handle incomplete data for a collection of trees.

The basic idea is first to convert each tree $T \in \Psi$ with incomplete observations to multiple sub-trees sharing the same source but with complete observations. For tree T with incomplete data in a collection of tree Ψ, assume that the outcomes of the kth probe sent by $\rho(T)$ are only observable by $R_k(T) \subseteq R(T)$. With probe k, we associate the multicast tree T_k that spans the root $\rho(T)$ and $R_k(T)$. This is obtained by finding the spanning tree of $\rho(T)$ and $R_k(T)$ in T. Therefore, the tree T with incomplete observation can be treated as a set of trees $\{T_k\}_{k=1,\ldots,n_T}$, each of which is with complete observation. Note that the same tree may appear many times in $\{T_k\}_{k=1,\ldots,n_T}$ and can be merged as one tree with multiple probes. For each tree with incomplete data in Ψ, we replace it with the set of its subtrees with complete data and add these trees

to Ψ. We then have a set of trees each of which has complete data and the algorithms described in Sections 4 and 6.1 can be applied to the inference of loss rate and delay distribution.

7. EM ALGORITHM FOR LOSS INFERENCE

We find convenient to work with the log-likelihood function

$$\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha) = \sum_{T\in\Psi} \mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha) \tag{15}$$

where
$\mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha) = \sum_{x_{R(T)}\in\Omega_{R(T)}}(n_T(x_{R(T)})\log p(x_{R(T)};\alpha))$ is the log-likelihood of the the measurement down the tree $T\in\Psi$. We estimate α by the maximizer of the likelihood (15), namely $\widehat{\alpha} = \arg\max \mathcal{L}^{\rm inc}{}_T(\mathbf{x}_{R(T)};\alpha)$. We follow the approach in [7, 8] and employ the EM algorithm to obtain an iterative approximation to the maximizer of (15). The basic idea is that rather than performing a complicated maximization, we "augment" the observed data with *unobserved* or *latent* data so that the resulting log-likelihood has a simpler form. Following [8], we augment the actual observations with the *unobserved* observations at the interior links. In other words, we assume complete knowledge of the loss process. The log-likelihood for the *complete data* $\mathbf{x} = (\mathbf{x}_T)_{T\in\Psi}$ is

$$\mathcal{L}(\mathbf{x};\alpha) = \sum_{T\in\Psi}\mathcal{L}(\mathbf{x}_T;\alpha) \tag{16}$$

where $\mathcal{L}(\mathbf{x}_T;\alpha) = \log p(\mathbf{x}_T;\alpha)$ is the log-likelihood of the complete set data for T. It is easy to realize that $p(x_T^1,\ldots,x_T^{n_T};\alpha) = \prod_{(i,j)\in E(T)}\alpha_{i,j}^{n_{j,T}}\overline{\alpha}_{i,j}^{n_{i,T}-n_{j,T}}$ and that

$$\begin{aligned}\mathcal{L}(\mathbf{x};\alpha) &= \sum_{(i,j)\in E(\Psi)}\Big(\sum_{T\in\Psi_{i,j}} n_{j,T}\log\alpha_{i,j} \\ &\quad + \Big(\sum_{T\in\Psi_{i,j}} n_{i,T} - \sum_{T\in\Psi_{i,j}} n_{j,T}\Big)\log\overline{\alpha}_{i,j}\Big).\end{aligned} \tag{17}$$

Maximization of (17) is trivial, as the stationary point conditions

$$\frac{\partial\mathcal{L}(\mathbf{x};\alpha)}{\partial\alpha_{i,j}} = 0 \quad (i,j)\in E(\Psi) \tag{18}$$

immediately yield

$$\widehat{\alpha}_{i,j} = \frac{\sum_{T\in\Psi_{i,j}} n_{j,T}}{\sum_{T\in\Psi_{i,j}} n_{i,T}} \quad (i,j)\in E(\Psi). \tag{19}$$

Since $\mathbf{x}$ and thus the counts except for leaves are not known, the EM algorithm uses the complete log-likelihood $\mathcal{L}(\mathbf{x};\alpha)$ to iteratively find $\widehat{\alpha}$ as follows:

1. *Initialization.* Select the initial link loss rate $\widehat{\alpha}^{(0)}$. The simulation study suggests the values that the algorithm converges to are independent of initial values.

2. *Expectation.* Given the current estimate $\widehat{\alpha}^{(\ell)}$, compute the conditional expectation of the log-likelihood given the observed data $\mathbf{x}$ under the probability law induced by $\widehat{\alpha}^{(\ell)}$,

$$\begin{aligned}Q(\alpha';\widehat{\alpha}^{(\ell)}) &= E_{\widehat{\alpha}^{(\ell)}}[\mathcal{L}(\mathbf{x};\alpha')|\mathbf{x}_R] \\ &= \sum_{(i,j)\in E(\Psi)}\Bigg\{\sum_{T\in\Psi_{i,j}}\widehat{n}_{j,T}\log\alpha'_{i,j} \\ &\quad + \Big(\sum_{T\in\Psi_{i,j}}\widehat{n}_{i,T} - \sum_{T\in\Psi_{i,j}}\widehat{n}_{j,T}\Big)\log\overline{\alpha}'_{i,j}\Big)\Bigg\}\end{aligned} \tag{20}$$

where $\hat{n}_{k,T} = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n_{k,T}|\mathbf{x}_R]$. $Q(\alpha';\widehat{\alpha}^{(\ell)})$ has the same expression as $\mathcal{L}(\mathbf{x};\alpha')$ but with the actual *unobserved* counts $n_{k,T}$ replaced by their conditional expectations $\widehat{n}_{k,T}$. To compute $\hat{n}_{k,T}$, remember that $n_{k,T} = \sum_{m=1}^{n_T} x_{k,T}^m$. Thus, we have

$$\begin{aligned}\widehat{n}_{k,T} &= \sum_{m=1}^{n_T}\mathsf{P}_{\widehat{\alpha}^{(\ell)}}[X_{k,T}=1|X_{R(T)}=x_{R(T)}^m] \\ &= \sum_{x_{R(T)}\in\Omega_{R(T)}} n_T(x_{R(T)})\mathsf{P}_{\widehat{\alpha}^{(\ell)}}[X_{k,T}=1|X_{R(T)}=x_{R(T)}]\end{aligned} \tag{21}$$

3. *Maximization.* Find the maximizer of the conditional expectation $\alpha^{(\ell+1)} = \arg\max_{\alpha'} Q(\alpha',\widehat{\alpha}^{(\ell)})$. The maximizer is given by (19) with the conditional expectation $\hat{n}_{k,T}$ in place of $n_{k,T}$.

4. *Iteration.* Iterate steps 2 and 3 until some termination criterion is satisfied. Set $\widehat{\alpha} = \widehat{\alpha}^{(\ell)}$, where ℓ is the terminal number of iterations.

Complexity
The complexity of the algorithm is dominated by computation of the conditional expectation $\widehat{n}_{k,T}$. This can be accomplished in linear time with $\#V(T)$, $T\in\Psi$. The algorithm is described in [3].

Convergence
We establish conditions for convergence of estimated parameters and likelihood under the EM algorithm for loss inference. Observe that the complete data log-likelihood function (17) can be written

$$\mathcal{L}(\mathbf{x};\alpha) = \sum_{T\in\Psi}\sum_{i\in V(T)\setminus\{\rho(T)\}} n_{i,T}\phi_{i,T}(\alpha) \tag{22}$$

where

$$e^{\phi_{i,T}(\alpha)} = \frac{\alpha_{f(i,T),i}}{\overline{\alpha}_{f(i,T),i}}\prod_{j\in d(i,T)}\overline{\alpha}_{i,j} \tag{23}$$

(Here the empty product when $d(i,T)=\emptyset$ is taken as 1). Thus the log likelihood comes from an exponential family with sufficient statistics $(n_{i,T})_{T\in\Psi,\ i\in V(T)}$ and parameters α. The exponential family is *regular*, since we take α in the convex set $\mathcal{A} = (0,1)^{\times_{T\in\Psi}V(T)}$. Note that the map $\alpha\mapsto\phi$ is invertible: $e^{\phi_{i,T}} = \alpha_{f(i,T),i}/\overline{\alpha}_{f(i,T),i}$ for a receiver i in $R(T)$. Invertibility then follows by induction: if we know all the $(\alpha_{i,j})_{j\in d(i,T)}$ then we can recover $\alpha_{f(i,T),i}$ from ϕ_i. It follows that the exponential family is *curved*: the $\phi_{i,T}$ are constrained to some $\#V$-dimensional smooth submanifold of $\mathbb{R}^{\times_{T\in\Psi}V(T)\setminus\{\rho(T)\}}$ through the constraint that the link probabilities α calculated from ϕ_T on different trees T must agree on common links.

The following convergence results for the sequence of EM iterates $\hat{\alpha}^{(\ell)}$ follow from the regular exponential family property; see Theorem 6 in [20].

THEOREM 3. *(i) $\mathcal{L}^{\rm inc}(\mathbf{x}_R;\hat{\alpha}^{(\ell)})$ converges to some limit L.*

(ii) If $\{\alpha\in\mathcal{A}\mid\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)=L\}$ is discrete, $\hat{\alpha}^{(\ell)}$ converges to some α^* that is a stationary point of $\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)$.

(iii) If $\mathcal{L}i(\mathbf{x}_R;\alpha)$ is unimodal, $\hat{\alpha}^{(\ell)}$ converges to the incomplete data MLE, namely, $\hat{\alpha} = \arg\max_\alpha\mathcal{L}^{\rm inc}(\mathbf{x}_R;\alpha)$

The theorem implies that when there are multiple stationary points, *e.g.* local maxima, the EM iterates may not converge to the global maximizer. Unfortunately, we were not able to establish whether there is a unique stationary point or conditions under which unicity holds.

8. SUMMARY

In this paper, we focused on inferring network internal link-level performance from end-to-end multicast measurements taken from a collection of trees. We addressed two questions:

- Given a collection of multicast trees, whether all of the links (or a specified subset) are identifiable.
- If a set of links of interest are identifiable, how do we obtain accurate estimates of their performance.

With loss rates as performance metrics, we established necessary and sufficient conditions for identifiability; and proposed two algorithms, MVWA algorithm and EM algorithm for inferring a set of links of interests. The algorithms are evaluated through model simulation and network simulation. The model simulation suggests that the EM algorithm is more accurate and of less variability. In the network simulation, we observe that EM algorithm can provide accurate estimate to the probe traffic loss whereas over-estimate all traffic loss slightly. Moreover, we extend the EM algorithm infer link delays, and demonstrate how to use our algorithms when only unicast measurement are available or some of the observations made at end-hosts are missing.

9. REFERENCES

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, and D. Towsley. "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", *IEEE Communications Magazine*, May 2000.

[2] M. Adler, T. Bu, R. Sitaraman, and D. Towsley. "Tree Layout for Internal Network Characterizations in Multicast Networks", *Proc. NGC'01*, London, UK, Nov. 2001.

[3] T. Bu, N.G. Duffield, F. Lo Presti, and D. Towsley. "Network Tomography on General Topologies". *UMass CMPSCI Technique Report.*

[4] R. Cáceres, N.G. Duffield, J.Horowitz, and D. Towsley. "Multicast-Based Inference of Network Internal Loss Characteristics" IEEE Trans. on Information Theory, vol. 45, pp. 2462-2480, 1999.

[5] R. Caceres, N.G. Duffield, J. Horowitz, D. Towsley, and T. Bu. "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation". *Proceedings of INFOCOM'99.*

[6] M. Coates and R. Nowak. "Network loss inference using unicast end-to-end measurement", *Proc. ITC Conf. IP Traffic, Modeling and Management*, Monterey, CA, September 2000.

[7] M. Coates and R. Nowak. "Sequential Monte Carlo Inference of Internal Delays in Nonstationary Communication Networks," submitted for publication, Jan 2001.

[8] M.J. Coates and R. Nowak. "Network Delay Distribution Inference from End-to-end Unicast Measurement," Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, May 2001.

[9] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley. "Network Delay Tomography from End-to-End Unicast Measurements", Proc. of the 2001 International Workshop on Digital Communications 2001 Evolutionary Trends of the Internet, Taormina, Italy, September 2001.

[10] N.G. Duffield, J. Horowitz, D. Towsley, W. Wei, and T. Friedman. "Multicast-based loss inference with missing data", to appear in IEEE Journal of Selected Areas in Communications

[11] N.G. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. "Inferring Link Loss Using Striped Unicast Probes", Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[12] Omer Gurewitz and Moshe Sidi. "Estimating One-way Delays from Cyclic-Path Delay Measurements", Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[13] Khaled Harfoush, Azer Bestavros, and John Byers. "Robust Identification of Shared Losses Using End-to-End Unicast Probes", Proc. IEEE ICNP 2000, Osaka, Japan.

[14] K. Lai and M. Baker. "Measuring link bandwidths using a deterministic model of packet delay," *Proc. SIGCOMM 2000*, Sweden, August 2000.

[15] Geoffrey J. McLachlan and Thriyambakam Krishnan. The EM algorithm and extensions. John Wiley, New York (1997)

[16] R. Penrose. "A Generalized Inverse for Matrices." Proc. Cambridge Phil. Soc. 51, 406-413, 1955.

[17] F. Lo Presti, N.G. Duffield, J.Horowitz, and D. Towsley. "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[18] `ns` – Network Simulator. See http://www-mash.cs.berkeley.edu/ns/ns.html

[19] Y. Shavitt, X. Sun, A. Wool, and B. Yener. "Computing the unmeasured: an algebraic approach to mapping the Internet," Proc. IEEE INFOCOM 2001, Anchorage, AK, April 2001.

[20] C.F. Jeff Wu. "On the convergence properties of the EM algorithm", Annals of Statistics, vol. 11, pp. 95-103, 1982.

[21] Abilene Network Operations Center. http://www.abilene.iu.edu/

[22] The Abilene network multicast deployment. http://www.abilene.iu.edu/images/ab-mcast.pdf

Multicast-Based Loss Inference with Missing Data*

Nick Duffield[‡] Joseph Horowitz[♠] Don Towsley[§] Wei Wei[§] Timur Friedman[§]

[‡]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

[♠]Dept. Math. & Statistics
University of Massachusetts
Amherst, MA 01003, USA
joeh@math.umass.edu

[§]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
{towsley,weiwei,friedman}@cs.umass.edu

Abstract

Network tomography using multicast probes enables inference of loss characteristics of internal network links from reports of end-to-end loss seen at multicast receivers. In this paper we develop estimators for internal loss rates when reports are not available on some probes or from some receivers. This problem is motivated by the use of unreliable transport control protocols, such as RTCP, to transmit loss reports to a collector for inference. We use a maximum likelihood (ML) approach in which we apply the Expectation Maximization (EM) algorithm to provide an approximate value for the ML estimator for the incomplete data problem. We present a concrete implementation of the algorithm that can be applied to measured data. For certain classes of models we establish identifiability of the probe and report loss parameters, and convergence of the EM sequence to the MLE. Numerical results suggest that these properties hold more generally. We derive convergence rates for the EM iterates, and the estimation error of the MLE. Last, we evaluate the accuracy and convergence rate through extensive simulations.

Keywords: End-to-end Measurement, Network Tomography, Missing Data, Maximum Likelihood Estimation, EM Algorithm, Multicast, RTP, RTCP.

1 Introduction

1.1 Motivation

As the Internet grows in size and diversity, its internal performance becomes ever more difficult to measure. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data.

One promising technique that avoids these problems, *Multicast Inference of Network Characteristics* (MINC), uses end-to-end multicast measurements to infer link-level loss rates and delay statistics by exploiting the inherent (and well characterized) correlation in performance observed by multicast receivers. These measurements do not rely on administrative access to internal nodes since the inference can be calculated using only information recorded at the end hosts.

The key intuition for inferring packet loss is that the arrival of a packet at a given internal node can be directly inferred from the packet's arrival at one or more receivers reached from the source by paths through that node; if it reaches the receivers, it must have reached the node. Conditioning on arrival at

*This work was supported in part by DARPA and the AFL under agreement F30602-98-2-0238

a descendent, we can determine the probability of successful transmission to and beyond the given node. Efficient inference algorithms are given in [2] for loss, [15] for delay distributions, [9] for delay variances, and [3] for inferring the logical multicast tree topology itself. Extensions of these ideas to unicast (where multicast is replaced by a packet pair [5] or a packet stripe [10]) have also been proposed.

All of the algorithms based on the MINC methodology rely on the availability of complete information from the receivers. This poses a serious problem in their deployment. For example, one promising avenue of deployment is through the extension of RTCP, the RTP [20] control protocol, to provide extended loss reports [11]. By piggybacking MINC loss reports on a standard transport protocol, one can effectively co-opt regular applications and their traffic to form a lightweight impromptu measurement infrastructure that encompasses many host end-points. However, loss reports are typically transmitted unreliably. Furthermore, the RTP standard imposes a constraint on the bandwidth that can be used by RTCP packets. Thus, this deployment will result in the availability of only *incomplete data sets* for the purpose of network inference. The need to analyze incomplete data sets also arises in the extension of the MINC techniques to unicast as they rely on collecting data from subsets of receivers. Thus there is a need to modify the inferencing techniques to be able to handle incomplete data sets. Using loss as an example, the goal of this paper is to extend the techniques developed in [2] to handle incomplete data.

1.2 Contribution

In this paper we adapt the multicast inference techniques proposed in [2] to perform inference of internal network characteristics when data is missing from some of the receivers. The data for the inference comprises measured end-to-end loss of multicast probes sent from a source to a number of destinations but where only a subset of the destinations report their observations for each multicast probe. This is used to infer the loss characteristics of each logical link of the tree joining the source to the destinations, i.e., of the composite paths between its branch points.

A simple approach to manage the impact of missing data would be to restrict inference to subsets of probes and receivers for which complete data is available, then patch together such estimators to draw inference on the complete tree. There are three drawbacks with this approach: (i) unless the coverage is sufficiently rich, it is not possible to infer transmission probabilities for all links; (ii) unless the missing data distribution obeys certain conditions–known as Missing Completely at Random (MCAR)–such estimators are not consistent in that they remain biased even in the limit of infinitely many probes; and (iii) even under MCAR such estimators are not generally efficient, i.e., there can exist estimators with smaller variance.

For these reasons we follow a more direct approach. We extend the Maximum Likelihood (ML) formulation of [2] to include the occurrence of missing data. The link loss probabilities are then estimated by the Maximum Likelihood Estimator (MLE) arising from the corresponding likelihood function. In contrast to the results in [2], it is not generally possible to determine the MLE by simple root finding when data is missing. Instead, we use the Expectation Maximization (EM) algorithm [7] to generate an approximating sequence to the corresponding MLE. We now outline the remainder of the paper and the detailed contributions.

In Section 2 we set up models for the multicast tree, probe propagation, and report loss, and review some

results for loss inference from complete data from [2]. We describe the model frameworks for missing data and give two examples: inference using unicast stripes [10], and inference using extended RTCP reports, as proposed in [1]. In Section 3 we set up the incomplete data likelihood function, and describe the EM algorithm and its application to the present model. We establish conditions required for convergence of the EM iterates to the MLE. We translate these into conditions on the measured data. If these conditions are not fulfilled, it is possible to pass instead to one or more related inference problems on subtrees for which the conditions are fulfilled. In Section 4 we tailor the EM algorithm to our specific problem and present an algorithm for use on measured data. Section 5 addresses conditions for identifiability of model parameters and relates these to topological properties of families of subtrees on which complete measurements can be made. Convergence of the MLE as the number of probes grows is investigated in Section 6; in particular we obtain explicit expressions for the asymptotic variance of the MLE for a class of simple models. A related expression for the convergence rate of the EM iterates is obtained in Section 7. The algorithm from Section 4 is evaluated in model-based simulation and using experimentally derived traces in Section 8. We conclude in Section 9. Details of most proofs are deferred to Section 10.

1.3 Related work

Several tools and methodologies exist for characterizing link-level behavior from end-to-end multicast measurements. However, most of these require complete data from all of the receivers in the multicast tree. These include the MINC methodologies for losses, [2], and for delay, [15, 9] and topology characteristics, [3]. These methodologies have been adapted to unicast through the transmission of packet pairs [5] or stripes [10] to pairs of receivers within a distribution tree. The data then consists of observations from pairs of receivers and can be interpreted as observations in which the data is missing from all but these pairs of receivers. The methodology presented in [10] treats the problem as separate problems corresponding to each pair of receivers and produces link estimates by averaging over all of the estimates produced from each of these receiver pair problems.

In [5], the authors introduce an additional link parameter, namely the conditional probability that the second packet within a pair is not lost given that the first packet is not lost. The authors then treat the outcomes of the each of the packets in a pair within the tree as unobserved data and use the EM algorithm to infer the link probabilities and conditional link probabilities. Due to the complexity of this task, they propose a heuristic for inferring these parameters. Because we rely on multicast, our task is simplified as we only have one set of link parameters to infer. Our solution methodology uses the EM algorithm to obtain a solution to the likelihood equation. Coates and Nowak have extended their EM-based, unicast-based techniques to infer delay statistics in [6].

Last, there exist several approaches that infer round trip link behavior. These include `pathchar` [8, 12] and the linear algebraic approach of [21]. The former infers loss, delay, and available link bandwidth whereas the latter infers round trip link delays. The former requires considerable time to converge. Both lose accuracy with asymmetric round trip paths.

2 Models for Probe and Report Transmission

2.1 Tree model

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify one node, the root ρ, with the source of probes, and set of leaves $R \subset V$ with the set of receivers. We assume that the root has single child, denoted by 1. If not, then we can treat separately the trees descended through each child of ρ, each one having this property. Each node k, apart from the root ρ, has a parent $f(k)$ such that $(f(k), k)$ is a link in L. We will sometimes refer to the link $(f(k), k)$ that terminates at k simply as link k. Define recursively the ancestors of k by $f^n(k) = f(f^{n-1}(k))$ with $f^0(k) = k$. We say j is descended from k, and write $j \prec k$, if $k = f^n(j)$ for some $n \in \mathbb{N}$. The set of children of k, namely $\{j \in V : f(j) = k\}$ is denoted by $d(k)$. $\mathcal{T}(k) = (V(k), L(k))$ will denote the subtree rooted at k; $R(k) = R \cap V(k)$ is the set of receivers in $\mathcal{T}(k)$. Define $U = V \setminus \{\rho\}$.

2.2 Packet loss model

We assume a Bernoulli loss model in which probes are independent and each probe is successfully transmitted across link k with probability α_k. Thus the progress of each probe down the tree is described by an independent copy of a stochastic process $X = (X_k)_{k \in V}$ as follows. $X_\rho = 1$. $X_k = 1$ if the probe reaches node $k \in V$ and 0 otherwise. If $X_k = 0$, then $X_j = 0, \forall j \prec k$. Otherwise, $P[X_j = 1 | X_{f(j)} = 1] = \alpha_j$ and $P[X_j = 0 | X_{f(j)} = 1] = 1 - \alpha_j$. We adopt the convention $\alpha_\rho = 1$ and denote $\alpha = (\alpha_i)_{i \in V}$. P_α will denote the distribution of X.

2.3 Inference of link loss from complete data

When a probe is sent down the tree from the root ρ, we cannot observe the whole process X. We assume that, at most, we know only the outcome $(X_k)_{k \in R} \in \Omega = \{0, 1\}^R$ that indicates whether or not the probe reached each receiver. When the entire outcome for a probe is known (i.e. X_k for all receivers k), we shall say that we have complete data from that probe. In [2] it was shown how the link probabilities can be determined from the the distribution of (complete) outcomes. We briefly review this.

Consider an experiment in which n probes are dispatched from the root ρ. Each probe $i = 1, \ldots, n$ gives rise to an independent realization $X^{(i)}$ of the probe process X. We call

$$X_{\text{cplt}} = (X_k^{(i)})_{k \in R}^{i=1,\ldots,n} \tag{1}$$

the complete data for the experiment. For each outcome $x \in \Omega$, let $n(x)$ denote the number of probes $i = 1, \ldots n$ for which $X_k^{(i)} = x_k$ for all $k \in R$. Let

$$p_\alpha(x) = \mathsf{P}_\alpha[X_k = x_k, \ \forall k \in R] \tag{2}$$

denote the probability of an outcome $x \in \Omega$. The *complete* data log-likelihood to obtain the data $X_{\text{cplt}} = (X^{(1)}, \ldots, X^{(n)})$ can be written in terms of the $n(x)$ as

$$\mathcal{L}_c(\alpha) = \log \mathsf{P}_\alpha[X_{\text{cplt}}] = \sum_{x \in \Omega} n(x) \log p_\alpha(x). \tag{3}$$

We characterize the Maximum Likelihood Estimator (MLE) of α, namely, $\arg\max_\alpha \mathcal{L}(\alpha)$, as follows. For $k \in V$, let A_k be the probability that the probe reaches k. Thus $A_k = \prod_{j \succeq k} \alpha_k$, the product of the probabilities of successful transmission on each link between k and the root ρ. For each $k \in U$ set

$$\gamma_k = \mathsf{E}_\alpha[\vee_{j \in R(k)} X_j] \tag{4}$$

i.e., γ_k is the probability that a probe reaches at least one receiver descended from node k. Denote by $\widehat{\gamma}_k$ the corresponding empirical quantity, i.e., the proportion of the n probes that reach at least one leaf descended from k:

$$\widehat{\gamma}_k = n^{-1} \sum_{i=1}^{n} \vee_{j \in R(k)} X_j^{(i)}. \tag{5}$$

In what follows we consider α to lie in the open parameter set $\mathcal{A} = \{\alpha \mid \alpha_k \in (0,1), k \in U\}$. Some of the results of the following theorem also hold on subsets of the boundary of $\mathcal{A}$.

Theorem 1 ([2]) *Assume $\alpha \in \mathcal{A}$.*

(i) For each $k \in U$,

$$(1 - \gamma_k/A_k) = \prod_{j \in d(k)} (1 - \gamma_j/A_k), \tag{6}$$

with the convention that an empty product occurs when $k \in R$ is zero.

(ii) Let $\mathcal{G} = \{(\gamma_k)_{k \in U} : \gamma_k > 0 \, \forall k;\ \gamma_k < \sum_{j \in d(k)} \gamma_j \, \forall k \in U \setminus R\}$. For each $\gamma \in \mathcal{G}$ and $k \in U$, (6) has a unique solution $\mathcal{H}_k(\gamma)$ in the interval $(\gamma_k, 1)$.

(iii) When $\widehat{\gamma} \in \mathcal{G}$, the likelihood equation,

$$\frac{\partial \mathcal{L}_c}{\partial \alpha_k}(\alpha) = 0, \quad k \in U \tag{7}$$

has as a unique solution

$$\widehat{\alpha}_k = \mathcal{K}_k(\widehat{\gamma}) := \mathcal{H}_k(\widehat{\gamma}) / \mathcal{H}_{f(k)}(\widehat{\gamma}), k \in U. \tag{8}$$

(iv) With probability one, for sufficiently large n, both $\widehat{\alpha}$ and the MLE of α lie in $\mathcal{A}$, and are hence equal.

(v) The parameters α are identifiable, i.e., $\mathsf{P}_\alpha = \mathsf{P}_{\alpha'}$ for $\alpha, \alpha' \in \mathcal{A}$ implies $\alpha = \alpha'$.

It turns out that Theorem 1(iv) is weaker than required for the present paper. We now establish a stronger version that provides a test as to whether or not $\widehat{\alpha}_k$ is the MLE for finite n.

Theorem 2 *Assume $\widehat{\gamma} \in \mathcal{G}$. If $\widehat{\alpha} \in \mathcal{A}$, then it is the MLE for α.*

Remark: Theorem 1(iv) establishes that for n sufficiently large, the MLE lies in $\mathcal{A}$ and hence must be $\widehat{\alpha}$, the solution of the likelihood equation. Theorem 2 is more useful from the computational point of view; it says that provided $\widehat{\alpha}$ lies in $\mathcal{A}$, a condition that can be checked by inspection, it is the MLE, regardless of n.

As a consequence of the MLE property, $\widehat{\alpha}$ is consistent ($\widehat{\alpha} \stackrel{n \to \infty}{\longrightarrow} \alpha$ with probability 1), and asymptotically normal ($\sqrt{n}(\widehat{\alpha} - \alpha)$ converges in distribution to a multivariate Gaussian random variable as $n \to \infty$); see e.g. [19].

2.4 Missing data model

We now want to generalize the problem by admitting the possibility that some outcomes may not be completely known because the receiver variables are missing. Let $T = (T_k^{(i)}))_{k\in R}^{i=1,\ldots,n}$ denote the $n \times \#R$ matrix of missing data indicators, with $T_k^{(i)}$ taking the value 0 if the variable $X_k^{(i)}$ is missing, and $T_k^{(i)} = 1$ if it is present. The set of observed data and missing data are thus, respectively,

$$X_{\rm obs} = \{X_k^{(i)} \mid T_k^{(i)} = 1\} \quad \text{and} \quad X_{\rm mis} = \{X_k^{(i)} \mid T_k^{(i)} = 0\}. \tag{9}$$

In this paper we assume that the missing data mechanism is ignorable in a sense we now describe; see [14] for further details. We treat T as a random variable whose distribution is parameterized by some quantity θ. P_θ will denote the distribution of T under θ, and $P_{\alpha,\theta}$ the joint distribution of $X_{\rm cplt}$ and T. We henceforth assume that the missing data is *missing at random* (MAR). This is the property that the distribution of the missing-data mechanism T does not depend on the missing values $X_{\rm mis}$. More formally, we can write the MAR property as $\mathsf{P}_\theta[T \mid X_{\rm obs}, X_{\rm mis}] = \mathsf{P}_\theta[T \mid X_{\rm obs}]$. As a consequence of MAR it can be shown that the joint distribution of the observed data and the missing-data mechanism enjoys the following factorization property:

$$\mathsf{P}_{\theta,\alpha}[X_{\rm obs}, T] = \mathsf{P}_\theta[T \mid X_{\rm obs}]\mathsf{P}_\alpha[X_{\rm obs}]. \tag{10}$$

Assuming the parameters (α, θ) to be distinct with product parameter space $\mathcal{A} \times \Theta$, (10) says that the missing data mechanism is ignorable in that likelihood-based inference for α based on the joint likelihood $\mathsf{P}_{\theta,\alpha}[X_{\rm obs}, T]$ are the same as those based upon $\mathsf{P}_\alpha[X_{\rm obs}]$. Thus for purposes of inferring α, we may ignore the parameters θ of the missing data mechanism. A special case of MAR is data *missing completely at random* (MCAR). With MCAR the missingness probabilities do not depend on any data: $\mathsf{P}_\theta[T \mid X_{\rm obs}] = \mathsf{P}_\theta[T]$.

2.5 Examples

We describe two applications where data is missing and place them into the framework described above.

Inference using unicast data. In [10], the authors describe an approach to *unicast* based inference in which n sets of packets, known as stripes, are transmitted by a source to all receiver pairs. The motivation is that within each stripe, packets are transmitted back-to-back, and so their loss behavior on common links should be highly correlated. With perfect correlations (i.e. both packets being either transmitted or lost on a common link) the stripe has the same behavior as a notional multicast packet that follows the same route and is subject to the same loss.

We can put each receiver pair in correspondence with a missing data indicator as follows. $T^{(i)} = (T_k^{(i)})_{k\in R}$ identifies the pair of receivers corresponding to the i-th stripe, i.e., $T_j^{(i)} = T_k^{(i)} = 1$, $T_l^{(i)} = 0$, $l \in R, l \neq j, k$ if the pair of receivers is $j, k \in R$, $j \neq k$. Thus missingness of data from probe i at receiver ℓ occurs because ℓ is not a member of the pair of receiver nodes selected for the probe.

If the receiver pairs are chosen independently from stripe to stripe using the same distribution, then $T = (T^{(i)})_{i=1}^{n}$ is a sequence of IID random variables. Thus T has the following distribution,

$$\mathsf{P}[T = t] = \prod_{i=1}^{n} \mathsf{P}[T^{(i)} = t^{(i)}], \quad \forall t \in \{0,1\}^{\#R} \tag{11}$$

where

$$\mathsf{P}[T^{(i)} = t] = \sum_{j,k \in R; j \neq k} \mathbf{1}\{t_j = 1\}\mathbf{1}\{t_k = 1\} \prod_{l \in R \setminus \{j,k\}} \mathbf{1}\{t_l = 0\} p_{j,k}, \quad \forall t \in \{0,1\}^{\#R}$$

Here $p_{j,k}$ is the probability that the pair of receivers j and k is chosen. If we further assume that T is independent of X, then the data is MCAR.

Another variation has the sender cycle through the pairs in a round robin fashion. Let $\pi : R^2 \setminus \{(j,j) : j \in R\} \to \{1, \ldots, m\}$ be a one-to-one mapping where $\pi(j,k)$ is the position in the round robin schedule where a probe is sent to receiver pair j and $k$and $m = \#R \times (\#R - 1)$. The joint probability distribution for T is given by (11) with

$$\mathsf{P}[T^{(mi+d)} = t] = \sum_{j,k \in R; j \neq k} \mathbf{1}\{\pi(j,k) = d\}\mathbf{1}\{t_j = 1\}\mathbf{1}\{t_k = 1\} \prod_{l \in R \setminus \{j,k\}} \mathbf{1}\{t_l = 0\}, \tag{12}$$

for all $t \in \{0,1\}^{\#R}, i \geq 0, 1 \leq d \leq m$.

Inference using RTP/RTCP. The Reliable Transport Protocol (RTP) [20] is a protocol for the transfer of data from a single sender to one or more receivers. Associated with it is a control protocol RTCP that allows receivers to broadcast loss behavior to each other and to a third party. Typically, the observations are batched and each batch is broadcast as a single report. The third party can collect the observations and apply inference methodologies to them. However, these reports are typically not transmitted reliably. Consequently, the data collector must deal with missing information.

In the current implementation of RTCP, receivers broadcast only average loss rates. Extensions to the protocol, proposed in [1], enable receivers to report on the reception of individual packets. However, due to the constraints imposed on reporting volumes by RTCP, it may not be possible to report on every packet. The omission of certain reports to fulfill this constraint is thus an additional source of missingness.

We propose a simple model for this scenario. Consider receiver $j \in R$ that collects loss observations and sends them to a data collector. Let $\{\{A_{i,j}\}_{i=1}^{\infty}\}_{j \in R}$ be a set of random variables where $A_{i,j}$ is the number of observations placed in the i-th report by the j-th receiver. Let $\{\{C_j^{(k)}\}_{k=1}^{\infty}\}_{j \in R}$ be indicator random variables that represent the outcome of the transmission of the k-th loss report by receiver j to the data collector; it is received by the collector if $C_j^{(k)} = 1$ and lost otherwise. Define $\pi(i,j) = \min\{l : \sum_{k=1}^{l-1} A_{k,j} < i \leq \sum_{k=1}^{l} A_{k,j}\}$, i.e., $\pi(i,j)$ identifies which report the i-th observation from receiver j is placed in. Let $\{\{S_j^{(i)}\}_{i=1}^{\infty}\}_{j \in R}$ be a set of indicator random variables that represent whether probe i was actually selected for reporting from receiver j; it is selected if $S_j^{(i)} = 1$. Then the missing data indicator $T_j^{(i)}, i = 1, \ldots, n, j \in R$ can be expressed as $T_j^{(i)} = S_j^{(i)} C_j^{(\pi(i,j))}$.

Under strong simplifying assumptions, namely that the random variables A, S and C are independent of X, the model is MCAR. However we can posit a situation in which independence may not hold in practice. Suppose the collector lies at a node k in the multicast tree. Then the path for reports from receivers in $R \setminus R(k)$ to k intersects with the paths of probe packets from ρ to receivers in $R(k)$. Thus we may expect the missingness variables $\{T_j^{(i)} : j \in R \setminus R(k)\}$ to be correlated with the receiver state $\{X_j : j \in R(k)\}$. This is precisely the type of model allowed when data is MAR.

2.6 Approaches to the problem of missing data

It is tempting to reduce the problem of inference with missing data to a composite of known inferences by performing inference using subsets of probes for which reports reached leaf descendents of a given node. A simple approach to manage the impact of missing data is to restrict inference to subsets of probes and receivers for which complete data is available, then patch together such estimators to draw inference on the complete tree. A minimal way to do this would be to use only probes for which reports were received from all receivers. A more sophisticated approach is the following:

(a) For each $k \in R$, estimate $\widehat{A}_k = \widehat{\gamma}_k$ by the fraction of observed reports indicating probe reception.

(b) For each $k \in U \setminus R$ let $\mathcal{R}_k$ denote the set of subsets of $R(k)$ in which each member is the descendant of a different child of k. For each $r \in \mathcal{R}_k$, use only probes with reports received from all $j \in r$ to form the fractions $\widehat{\gamma}_k(r)$ and $\{\widehat{\gamma}_j(r)\}_{j \in r}$. Estimate $\widehat{A}_k = (\sum_{r \in \mathcal{R}_k} \mathcal{H}_k(\widehat{\gamma}(r))/\#\mathcal{R}_k$, i.e., averaging over the $r \in \mathcal{R}_k$.

(c) Estimate link transmission probabilities $\widehat{\alpha}_k = \widehat{A}_k / \widehat{A}_{f(k)}$.

However, such "patchwork" approaches have three pitfalls:

(i) Unless the coverage is sufficiently rich, it is not possible to infer transmission probabilities for all links. If not all nodes are branch points of some "complete data" subtree, it follows from one of our later results that one cannot infer the transmission probability for the link that terminates at that node. In the minimal case, there may be *no* probes for which reports are received from all probes.

(ii) Such estimators are not consistent unless data is MCAR; we illustrate with an example in Section 3.1. Furthermore, checking whether a given data set is consistent with the MCAR property may be a complex task since the number of consistency conditions that would have to be checked grows exponentially with the number of leaves in the tree.

(iii) Even under MCAR such estimators are not generally efficient, i.e., there can exist estimators with smaller variance.

For these reasons we instead extend the previous ML approach to cover the missing data case directly: under general conditions ML-estimators are consistent and efficient. This is the subject of the next section.

3 Estimation of Link Loss Rates with Incomplete Data

In this section we present the likelihood function $\mathcal{L}$ for the incomplete data. Determination of the corresponding ML estimator for the link probabilities turns out to be significantly more complex that in the complete data case. We turn to a standard iterative method, the EM algorithm, to derive an approximating sequence to the incomplete data MLE.

3.1 Description of incomplete data and the likelihood function

The corresponding *incomplete* data likelihood function is the marginal distribution function of the observed data; formally we write this as $\int P_\alpha[X_{\rm obs}, X_{\rm mis}]dX_{\rm mis}$. We now obtain an explicit expression. In order to represent both missing and observed data in a compact form, we extend the set of outcomes to the set $\Omega^* = \{0, 1, \mathsf{u}\}^R$, where u is used to denote that a given receiver datum is missing. $\mathsf{u}^* = (\mathsf{u}, \ldots, \mathsf{u}) \in \Omega^*$ will denote the outcome in which data is missing from all receivers. Let $t \in \{0, 1\}^R$ denote the generic vector of missing data indicator variables. With each such t and $x \in \Omega$ we then associate an element $x(t)$ of Ω^* through

$$x_k(t) = \begin{cases} \mathsf{u} & \text{if } t_k = 0 \\ x_k & \text{otherwise} \end{cases}, \; k \in R. \tag{13}$$

An inverse of the above map associates with $x^* \in \Omega^*$ its missing data indicator $t(x^*)$ by

$$t_k(x^*) = \begin{cases} 0 & \text{if } x_k^* = \mathsf{u} \\ 1 & \text{otherwise} \end{cases}, \; k \in R. \tag{14}$$

The set of complete outcomes x that can give rise to an incomplete outcome x^* is the set

$$\Omega(x^*) = \{x \in \Omega \mid x_k^* = x_k \Leftrightarrow t_k(x^*) = 1\}, \quad \text{and conversely} \quad \Omega^*(x) = \{x^* \in \Omega^* \mid x \in \Omega(x^*)\} \tag{15}$$

is the set of complete outcomes x^* that can be obtained from a complete outcome x. The equivalent conditions $x \in \Omega(x^*)$ and $x^* \in \Omega^*(x)$ can be rewritten as $x(t(x^*)) = x^*$.

The probability to record an incomplete outcome $X^{(i)}(T^{(i)}) = x^*$ is denoted

$$q_{\alpha,\theta}(x^*) = \mathsf{P}_{\alpha,\theta}[X^{(i)}(T^{(i)}) = x^*]. \tag{16}$$

Now $\{X^{(i)}(T^{(i)}) = x^*\} = \{X^{(i)} \in \Omega(x^*)\} \cap \{T^{(i)} = t(x^*)\}$. Using the MAR property (10) we factorize

$$q_{\alpha,\theta}(x^*) = p_\alpha^*(x^*)\theta(x^*) \tag{17}$$

where

$$p_\alpha^*(x^*) = \mathsf{P}_\alpha[X^{(i)} \in \Omega(x^*)] = \sum_{x \in \Omega(x^*)} p_\alpha(x) \quad \text{and} \quad \theta(x^*) = \mathsf{P}_\theta[T^{(i)} = t(x^*) \mid X^{(i)} \in \Omega(x^*)]. \tag{18}$$

Without loss of generality we have taken the missingness probabilities themselves as parameters θ. Note that by the MAR property, for any $x \in \Omega(x^*)$, $\mathsf{P}_\theta[T^{(i)} = t(x^*) \mid X^{(i)} \in \Omega(x^*)] = \mathsf{P}_\theta[T^{(i)} = t(x^*) \mid X^{(i)} = x]$. Since $\{t(x^*) \mid x^* \in \Omega^*(x)\} = \{0, 1\}^R$, the conditional probabilities θ satisfy

$$\sum_{x^* \in \Omega^*(x)} \theta(x^*) = 1, \quad \forall x \in \Omega. \tag{19}$$

Now let $m(x^*)$ denote the number of probes $i = 1, \ldots, n$ for which $X^{(i)}(T^{(i)}) = x^*$. Due to the factorization property (10), the log-likelihood function $\log \prod_{i=1}^{n} q_{\alpha,\theta}(X^{(i)}(T^{(i)}))$ can be written as a sum of

$$\mathcal{L}(\alpha) = \sum_{x^* \in \Omega^*} m(x^*) \log p^*_\alpha(x^*), \tag{20}$$

with a term that is independent of α. Thus, for the purposes of obtaining an ML estimate of α, we need only consider $\mathcal{L}(\alpha)$. We refer to $\mathcal{L}$ as the incomplete data likelihood function. Note that the term in $m(\mathsf{u}^*)$ makes no contribution to $\mathcal{L}$ since $\Omega(\mathsf{u}^*) = \Omega$ and hence $p^*_\alpha(\mathsf{u}^*) = 1$. Hence the sum in (20) can be restricted to $\Omega^*_0 = \Omega^* \setminus \{\mathsf{u}^*\}$.

Example: 2 leaf tree with MAR data. We now give an example to show how the complete data MLE, applied to those probes for which complete data is available, generally produces an inconsistent estimate of the link probabilities in the MAR case. Consider a two leaf tree where data is MAR from the right leaf; the probability of missingness thus depends on the data observed at the left leaf. The leaf probabilities obey:

$$q(11) = \alpha_1\alpha_2\alpha_3\theta(11), \quad q(10) = \alpha_1\alpha_2\overline{\alpha}_3\theta(10), \quad q(01) = \alpha_1\overline{\alpha}_2\alpha_3\theta(01), \tag{21}$$

$$q(00) = (\overline{\alpha}_1 + \alpha_1\overline{\alpha}_2\overline{\alpha}_3)\theta(00), \quad q(1\mathsf{u}) = \alpha_1\alpha_2\theta(1\mathsf{u}), \quad q(0\mathsf{u}) = (1 - \alpha_1\alpha_2)\theta(1\mathsf{u}) \tag{22}$$

Using the four instances of (19), namely,

$$x = 11: \ \theta(11) + \theta(1\mathsf{u}) = 1; \quad x = 10: \ \theta(10) + \theta(1\mathsf{u}) = 1; \tag{23}$$

$$x = 01: \ \theta(01) + \theta(0\mathsf{u}) = 1; \quad x = 00: \ \theta(00) + \theta(0\mathsf{u}) = 1; \tag{24}$$

(21) reduces to

$$q(11) = \alpha_1\alpha_2\alpha_3\theta(11), \quad q(10) = \alpha_1\alpha_2\overline{\alpha}_3\theta(11), \quad q(01) = \alpha_1\overline{\alpha}_2\alpha_3\theta(01), \tag{25}$$

$$q(00) = (\overline{\alpha}_1 + \alpha_1\overline{\alpha}_2\overline{\alpha}_3)\theta(01), \quad q(1\mathsf{u}) = \alpha_1\alpha_2\overline{\theta}(11), \quad q(0\mathsf{u}) = (1 - \alpha_1\alpha_2)\overline{\theta}(01) \tag{26}$$

Now, the complete data MLE based on the corresponding complete data empirical probabilities $\widehat{q}(11), \widehat{q}(10), \widehat{q}(01), \widehat{q}(00)$ is

$$\widehat{\alpha}_1 = \frac{(\widehat{q}(11) + \widehat{q}(10))(\widehat{q}(11) + \widehat{q}(01))}{\widehat{q}(11)(\widehat{q}(11) + \widehat{q}(10) + \widehat{q}(01) + \widehat{q}(00))}, \quad \widehat{\alpha}_2 = \frac{\widehat{q}(11)}{\widehat{q}(11) + \widehat{q}(01)}, \quad \widehat{\alpha}_3 = \frac{\widehat{q}(11)}{\widehat{q}(11) + \widehat{q}(10)} \tag{27}$$

In the MCAR case, all $\theta(x^*)$ appearing in (25) would be equal, and substituting q for $\widehat{q}$ in (27) one recovers $\alpha_1, \alpha_2, \alpha_3$: the estimator is consistent. But in the general MAR case one obtains only

$$\frac{\alpha_1(\alpha_2\theta(11) + \overline{\alpha}_2\theta(01))}{\alpha_1\alpha_2\theta(11) + (1 - \alpha_1\alpha_2)\theta(01)} \neq \alpha_1, \quad \frac{\alpha_2\theta(11)}{\alpha_2\theta(11) + \overline{\alpha}_2\theta(01)} \neq \alpha_2, \quad \alpha_3, \tag{28}$$

i.e. only estimation of α_3 is consistent.

3.2 Application of the EM algorithm

We can in principle estimate the link probabilities α by the incomplete data MLE $\check{\alpha} = \arg\max_{\alpha} \mathcal{L}(\alpha)$ in (20) calculated from the counts of incomplete outcomes $\boldsymbol{m} = \{m(x^*) : x^* \in \Omega_0^*\}$. However, we have been unable to obtain a direct solution to the incomplete-data likelihood equation. Instead, we employ a standard statistical method,the Expectation-Maximization (EM) algorithm [7], to obtain an iterative approximation $\widehat{\alpha}^{(\ell)},\ \ell = 0, 1, \ldots$ to a stationary value of the incomplete data likelihood. The algorithm comprises the following steps:

(i) *Initialization.* Pick some initial link probabilities $\alpha^{(0)}$. This could be done, e.g., by setting $\widehat{\alpha}^{(0)} = \widehat{\alpha}$, the complete data MLE determined from the counts of complete outcomes $\boldsymbol{m}$ if these are non-zero. When complete data is not available, we can use the fact (see the proof of Theorem 5 in [2]) that $\gamma_k = A_k + O(\|\overline{\alpha}\|^2)$ to approximate $\alpha_k = A_k/A_{f(k)} \approx \gamma_k/\gamma_{f(k)} = (1-\overline{\gamma}_k)/(1-\overline{\gamma}_{f(k)}) \approx 1+\gamma_k-\gamma_{f(k)}$. This suggests the initial value

$$\widehat{\alpha}_k^{(0)} = 1 + \widehat{\gamma}_k - \widehat{\gamma}_{f(k)}. \tag{29}$$

(ii) *Expectation.* For each $\widehat{\alpha}^{(\ell)}$ find the conditional expectation of the complete log-likelihood given the incomplete data $Q(\alpha', \widehat{\alpha}^{(\ell)}) = \mathsf{E}_{\widehat{\alpha}^{(\ell)}}[\mathcal{L}_c(\alpha') \mid \boldsymbol{m}]$.

(iii) *Maximization.* Find the maximizer of the condition expectation: $\widehat{\alpha}^{(\ell+1)} = \arg\max_{\alpha'} Q(\alpha', \widehat{\alpha}^{(\ell)})$

(iv) *Iteration.* Iterate steps (ii) and (iii) until some termination criterion is satisfied.

For $k \in V$, define the conditional probabilities for a probe to reach $R(k)$ as

$$\widehat{\gamma}_{k,\alpha} = \mathsf{E}_{\alpha}[\vee_{j \in R(k)} X_k \mid \boldsymbol{m}]. \tag{30}$$

For notational convenience we write the conditional probability $\widehat{\gamma}_{k,\widehat{\alpha}^{(\ell)}}$ derived from the iterate $\widehat{\alpha}^{(\ell)}$ as $\widehat{\gamma}_k^{(\ell)}$.

Theorem 3 *Assume $\widehat{\gamma}^{(\ell)} \in \mathcal{G}$. Then*

$$\alpha_k^{(\ell+1)} = \mathcal{K}_k(\widehat{\gamma}^{(\ell)}),\ k \in U \tag{31}$$

provided that $\mathcal{K}(\widehat{\gamma}^{(\ell)})$ lies in $\mathcal{A}$.

We now investigate the question of convergence of the iterates $\widehat{\alpha}^{(\ell)}$. Whereas the complete data likelihood function can be shown to derive from a standard exponential family (see the proof of Theorem 2), the incomplete data likelihood function derives only from a curved exponential family. Thus we cannot use results based on standard exponential families (see e.g. [22]) alone to conclude convergence of $\widehat{\alpha}^{(\ell)}$ to $\check{\alpha}$. We now establish conditions under which the sequence exists in $\mathcal{A}$ and converges to the MLE for the incomplete data problem.

Theorem 4 *Assume $\widehat{\gamma}^{(\ell)} \in \mathcal{G}$ and $\mathcal{K}(\widehat{\gamma}^{(\ell)}) \in \mathcal{A}$ for all ℓ.*

(i) $\mathcal{L}(\widehat{\alpha}^{(\ell)})$ converges to some limit L.

(ii) If $\{\alpha \in \mathcal{A} \mid \mathcal{L}(\alpha) = L\}$ *is discrete,* $\widehat{\alpha}^{(\ell)}$ *converges to some* α^* *that is a stationary point* $\mathcal{L}$*, i.e.* $\frac{\partial \mathcal{L}}{\partial \alpha}(\alpha^*) = 0$*.*

(iii) If $\mathcal{L}$ *is unimodal,* $\widehat{\alpha}^{(\ell)}$ *converges to the incomplete data MLE* $\check{\alpha}$*.*

3.3 Calculation of the EM iterates

An algorithm for the calculating $\mathcal{K}_k(\gamma)$ for a given $\gamma \in \mathcal{G}$ has been detailed in [2]. It remains then to provide an algorithm for the calculation of the $\widehat{\gamma}_\alpha$. Let $n_0 = n - m^*(\mathsf{u}^*)$ denote the number of probes for which the data is not entirely missing. Observe that

$$\widehat{\gamma}_{k,\alpha} = \sum_{x^* \in \Omega_0^*} \frac{m(x^*)}{n_0} \, \widehat{\gamma}_{k,\alpha}(x^*) \tag{32}$$

where

$$\widehat{\gamma}_{k,\alpha}(x^*) = \mathsf{E}_\alpha[\vee_{j \in R(k)} X_k | X^* = x^*]. \tag{33}$$

Let $R(k, x^*) = \{j \in R(k) \mid t_j(x^*) = 1\}$ denote the receivers descended from k from which data is observable. Let $h(k, x^*)$ denote the closest ancestor h of k for which a packet has been observed to reach at least one descendant leaf, i.e.,

$$h(k, x^*) = \inf_{j \succeq k} \{j : y_j^* = 1\}, \tag{34}$$

where $y_k^* = \vee^*_{j \in R(k)} x_j^*$. When $k \prec j$, let $d(j, k)$ denote that child of j that is an ancestor (or possibly equal to) k, i.e., $d(j, k) = \{i \in d(j) : i \succeq k\}$.

Theorem 5 *When* $y_k^* = 1$, $\widehat{\gamma}_{k,\alpha}(x^*) = 1$*. When* $y_k^* < 1$*,*

$$\widehat{\gamma}_{k,\alpha}(x^*) = \frac{c_k - b_k}{c_{d(h,k)}} \prod_{k \prec i \preceq d(k,h)} \left\{ \alpha_i \prod_{j \in d(i) \setminus d(i,k)} c_j \right\} \tag{35}$$

where $h = h(k, x^*)$*, and for* $k \preceq i \prec h$*,*

$$b_k = \mathsf{P}_\alpha[\vee_{j \in R(k)} X_j = 0 \mid X_{f(k)} = 1], \qquad c_k = \begin{cases} 1, & \text{if } R(k, x^*) = \emptyset \\ \mathsf{P}_\alpha[\vee_{j \in R(k, x^*)} X_j = 0 \mid X_{f(k)} = 1] & \text{otherwise} \end{cases}. \tag{36}$$

Remark: it was found in [2] that the problem of determining the α for a tree with complete data factors into the problem of solving a set of depth two tree inference problems, one for each node $k \in V \setminus R$. For each leaf k one constructs the logical tree with root ρ having single child k, and $d(k)$ leaf-children. Furthermore, for a general tree, the problem could be mapped onto that for a binary tree by the insertion of lossless links. However, this method cannot be applied when there is missing data. This is because the form (35) for $\widehat{\gamma}^{(\ell)}_{k,\alpha}$ includes variables from receivers other than those descended from k.

3.4 Topology and data conditions

Theorems 3 and 4 required the iterators $\widehat{\gamma}^{(\ell)}$ to lie in the domain $\mathcal{G}$. In this section we specify conditions on the data in order for these requirements to hold. In some cases where the conditions do not hold, it is possible adjust the problem by passing to one or more subtrees of the original tree, for which the conditions do hold. The requirements for Theorems 3 and 4 are then fulfilled: see Lemma 1 below. The conditions describes here are similar to those applied in the case of complete data in [2].

Non-identifiable subtrees. Order the elements of the set $\{0, 1, \mathsf{u}\}$ as $\mathsf{u} < 0 < 1$ and extend the usual maximum operator $\vee$ on $\{0, 1\}$ to an operation $\vee^*$ on $\{0, 1, \mathsf{u}\}$, respecting the order in an obvious manner. For a given realization (X, T) of the single probe and missing data process, define the quantities

$$Y_k^* = \vee_{j \in R(k)}^* X_j(T) \tag{37}$$

i.e., the extended maximum of $X_j(T)$ over all receivers j descended from k. Y_k^* takes the value u if all data from $R(k)$ on a given probe is missing, 1 if a probe was observed to reach at least one receiver in $R(k)$, and 0 otherwise. We first eliminate from consideration subtrees on which no data is missing but whose leaves were reached by no probes. For $k \in V$, let D_k denote the event that for some probe i, $X_j^{(i)}(T^{(i)}) \neq 0$ for some $j \in R(k)$. We will assume

$$D_k \text{ occurs for all } k \in V \tag{38}$$

If (38) does not hold, the following procedure reduces the inference problem to one for which it does. If D_k fails, we remove from further consideration the subtree $\mathcal{T}(k)$ rooted at k. If this pruning leaves the parent $f(k)$ with only one offspring j, the remaining tree is no longer a logical multicast tree. To make it so we remove the link $(f(k), j)$ and identify the nodes j and $f(k)$. The consequence is that we will only able to identify the characteristics of the composite link joining j to $f^2(j)$ of the original tree. Performing these operations for all k at which D_k fails, we obtain a tree for which (38) holds.

In general, it is not possible to attribute a transmission probability, even of zero, to individual links in $\mathcal{T}(k)$, since we cannot distinguish the link or links with zero transmission rate. An exception to this is when D_k fails for a leaf node k, but $D_{f(k)}$ holds at the parent node $f(k)$. In this case we may estimate $\widehat{\alpha}_k = 0$. Except in this case, we flag all $\{\alpha_j : j \preceq k\}$ as unknown.

Links with perfect transmission. Let D_k' denote the complement of the event $\{X_j^{(i)}(T^{(i)}) = 1, \forall j \in R(k), i = 1, \ldots, n\}$. When D_k' fails, lossless transmission is reported for all probes to all receivers in $R(k)$. The effect is to position $\mathcal{K}(\widehat{\gamma}_\alpha)$ on the boundary of $\mathcal{A}$, since it follows that $\mathcal{K}_j(\widehat{\gamma}_\alpha) = 1$ for all $j \in R(k)$. Although this is not a problem for computation, it takes us out of the domain of application of Theorems 2, 3 and 4. The formal application of these results requires that

$$D_k' \text{ holds for all } k \in V. \tag{39}$$

If (39) does not hold, the following procedure reduces the inference problem to a set of one or more for which it does. When D_k' fails, we set $\widehat{\alpha}_j = 1$ for all nodes $j \in V(k)$, and omit these nodes from further

consideration. We then spawn a set of separate inference problems by forming the set of subtrees not containing k that are rooted at ancestors of k. This is the set of trees $\{\mathcal{T}(j,\ell) \mid j \succ k;\ \ell \in d(j), \ell \not\succ k\}$, where $\mathcal{T}(j,\ell)$ has vertices $\{j\} \cup V(\ell)$ and links $\{(j,\ell)\} \cup L(\ell)$.

Model Conformance. We also need a condition to ensure that estimated quantities $\widehat{\gamma}$ lie in $\mathcal{G}$. Let D_k' be the event that k has children $j, \ell \in d(k)$ such that $X_j^{(i)}(T^{(i)}) = 1$ and $X_\ell^{(i)}(T^{(i)}) \neq 0$. Without this condition, probe losses on different subtrees descended from k, conditional on the probe having reached k, are correlated. This is because each probe is observed on no more than one such subtree. Henceforth we assume

$$D_k'' \text{ holds for all any } k \in V \setminus R. \tag{40}$$

If D_k'' fails, we adjust the tree by removing the link $(f(k), k)$ from the tree and identifying its endpoints k and $f(k)$. In the original tree, we will only be able to identify the characteristics of the composite links joining $f(k)$ to the children $d(k)$. The procedure is iterated if necessary until (40) holds.

Conditions (38) and (40) enable us to fulfill some assumptions in Theorems 3 and 4. We will henceforth assume that they hold.

Lemma 1 *When (38) and (40) hold, $\widehat{\gamma}_\alpha \in \mathcal{G}$ for any $\alpha \in \mathcal{A}$.*

3.5 Example: the two-receiver tree

In the simplest case we can establish unimodality of $\mathcal{L}_c$ directly, and thus conclude convergence of the EM iterates to the incomplete data MLE. Consider the two receiver tree with root ρ having a single child 1 whose children are the leaf nodes 2 and 3. In the two receiver tree, we enumerate $\Omega = \{11, 01, 10, 00\}$ and $\Omega_0^* = \{11, 01, 10, 00, 1\mathsf{u}, \mathsf{u}1, 0\mathsf{u}, \mathsf{u}0\}$. It is not difficult to determine that the $\widehat{\gamma}_{k,\alpha}(x^*)$ are as given by the following table:

x^*	$\widehat{\gamma}_{1,\alpha}(x^*)$	$\widehat{\gamma}_{2,\alpha}(x^*)$	$\widehat{\gamma}_{3,\alpha}(x^*)$
11	1	1	1
10	1	1	0
01	1	0	1
00	0	0	0
1u	1	1	α_3
u1	1	α_2	1
0u	$\frac{\alpha_1\overline{\alpha}_2\alpha_3}{\overline{\alpha}_1+\alpha_1\overline{\alpha}_2}$	0	$\frac{\alpha_1\overline{\alpha}_2\alpha_3}{\overline{\alpha}_1+\alpha_1\overline{\alpha}_2}$
u0	$\frac{\alpha_1\alpha_2\overline{\alpha}_3}{\overline{\alpha}_1+\alpha_1\overline{\alpha}_3}$	$\frac{\alpha_1\alpha_2\overline{\alpha}_3}{\overline{\alpha}_1+\alpha_1\overline{\alpha}_3}$	0

These yield

$$\begin{aligned}
n_0\widehat{\gamma}_{2,\alpha} &= m(11) + m(10) + m(1\mathsf{u}) + m(\mathsf{u}1)\alpha_2 + m(\mathsf{u}0)\frac{\alpha_1\alpha_2\overline{\alpha}_3}{\overline{\alpha}_1 + \alpha_1\overline{\alpha}_3} \\
n_0\widehat{\gamma}_{3,\alpha} &= m(11) + m(01) + m(\mathsf{u}1) + m(1\mathsf{u})\alpha_3 + m(0\mathsf{u})\frac{\alpha_1\overline{\alpha}_2\alpha_3}{\overline{\alpha}_1 + \alpha_1\overline{\alpha}_2} \\
n_0(\widehat{\gamma}_{2,\alpha} + \widehat{\gamma}_{3,\alpha} - \widehat{\gamma}_{1,\alpha}) &= m(11) + m(1\mathsf{u})\alpha_3 + m(\mathsf{u}1)\alpha_2
\end{aligned}$$

The EM iterates $(\mathcal{K}_1(\widehat{\gamma}_\alpha), \mathcal{K}_2(\widehat{\gamma}_\alpha), \mathcal{K}_3(\widehat{\gamma}_\alpha))$ of $(\alpha_1, \alpha_2, \alpha_3)$ are then

$$\mathcal{K}_1(\widehat{\gamma}_\alpha) = \frac{\widehat{\gamma}_{2,\alpha}\widehat{\gamma}_{3,\alpha}}{\widehat{\gamma}_{2,\alpha} + \widehat{\gamma}_{3,\alpha} - \widehat{\gamma}_{1,\alpha}}, \quad \mathcal{K}_2(\widehat{\gamma}_\alpha) = \frac{\widehat{\gamma}_{2,\alpha} + \widehat{\gamma}_{3,\alpha} - \widehat{\gamma}_{1,\alpha}}{\widehat{\gamma}_{3,\alpha}}, \quad \mathcal{K}_3(\widehat{\gamma}_\alpha) = \frac{\widehat{\gamma}_{2,\alpha} + \widehat{\gamma}_{3,\alpha} - \widehat{\gamma}_{1,\alpha}}{\widehat{\gamma}_{2,\alpha}}$$

Theorem 6 *In the two-receiver tree, the incomplete data likelihood function $\mathcal{L}$ is unimodal, and hence $\widehat{\alpha}^{(\ell)}$ converges to the incomplete data MLE provided that $\mathcal{K}(\widehat{\gamma}^{(\ell)}) \in \mathcal{A}$ for all ℓ.*

4 Network Inference Algorithm

In order to carry out inference on measured data, we express the calculation of $\widehat{\gamma}$ in Theorem 5 as an algorithm. We start by constructing b_k, c_k and y_k^* recursively. Clearly the b_k satisfy

$$b_k = \begin{cases} \overline{\alpha}_k, & k \in R, \\ \overline{\alpha}_k + \alpha_k \prod_{j \in d(k)} b_j, & k \in U \setminus R \end{cases} \tag{41}$$

The c_k satisfy a similar recursion:

$$c_k = \begin{cases} 1, & k \in R,\ x_k^* = \mathsf{u} \\ \overline{\alpha}_k, & k \in R,\ x_k^* \neq \mathsf{u} \\ \overline{\alpha}_k + \alpha_k \prod_{j \in d(k)} c_j, & k \in U \setminus R \end{cases} \tag{42}$$

The y_k^* satisfy the recursion

$$y_k^* = \begin{cases} x_k^*, & k \in R \\ \bigvee_{j \in d(k)}^* y_j^*, & k \in V \setminus R \end{cases} \tag{43}$$

We formally specify an algorithm for the calculation of the $\widehat{\gamma}_{k,\alpha}$ in Figure 1. The main procedure comprises two phases. In the first phase, `set_ybc`, calculates the y_k^*, b_k and c_k passing up the tree from the leaves. The second phase, `set_g`, then calculates the $\widehat{\gamma}_{k,\alpha}$ traversing the tree from the root ρ downwards. h_k plays the role of $d(k,h)$ while e plays the role of $\alpha_i \prod_{j \in d(i) \setminus \{d(i,k)\}} c_j$. On a given path down the tree, $\mathsf{flag} = 1$ until a node k with $y_k^* = 0$ is first encountered. $\mathsf{flag} = 0$ on all calls to `set_g` below k. The identity of the node $h(i, x^*)$ is then maintained in calls at nodes i below the child j of k (lines 10–13).

We note there is some redundancy in the algorithms, which can be avoided in implementations. b_k and c_k need not be calculated at nodes k for which $y_k^* = 1$, since these values are not used. The c_k depend only on the missing data indicator $t(x^*)$, and so need be calculated once for each set incomplete outcomes $\{x^* \in \Omega^* : t(x^*) = t\}$ having the same missing data indicator t. The b_k do not depend on x^*, and so may be calculated once in advance; in particular $b_k = c_k$ when x^* has no missing data, i.e., when $x_k^* \neq \mathsf{u}\ \forall k \in R$. Lastly, the y_k^* need only be calculated once for each probe with distinct x^*, and once at the start of the sequence of iterations.

5 Identifiability and Missing Data

We address the question of identifiability, i.e., whether there exists a unique set of model parameters giving rise to a given distribution of observable data. The multicast inference method exploits correlations between end-to-end measurements on intersection paths. Conversely, we expect that if the sets of receivers on which data from a given probe is observable are insufficiently rich, it will not be possible to infer the loss rates on all links. We give below a simple example that demonstrates this. In this section we shall derive conditions–between the topology and the subsets at which data is observable–that must be satisfied in order that the model parameters can be identified.

```
1.  procedure main(T, α, x*)
2.  set_ybc(T, α, x*, ρ);
3.  set_g(T, α, 1, ρ, ρ, 1);
4.  return({g_k : k ∈ V});
```

```
1.  procedure set_ybc(T, α, x*, k)
2.  if(d(k) == ∅) then {
3.      y*_k := x*_k;
4.      b_k := ᾱ_k;
5.      if(x*_k == u) then {c_k := 1;}
6.      else{c_k := ᾱ_k;}
7.  }
8.  else{
9.      foreach(j ∈ d(k) {
10.         (y*_j, b_j, c_j) := set_ybc(T, α, x*, j);
11.     }
12.     y*_k := ∨*_j y*_j;
13.     b_k := ᾱ_k + α_k ∏_{j∈d(k)} b_j;
14.     c_k := ᾱ_k + α_k ∏_{j∈d(k)} c_j;
15. }
16. return(y*_k, b_k, c_k);
```

```
1.  procedure set_g(T, α, e, k, h, flag)
2.
3.  if(y*_k == 1) then {
4.      g_k := 1;
5.      foreach(j ∈ d(k)){
6.          set_g(T, α, 1, j, ρ, 1);
7.      }
8.  }
9.
10. else{
11.     if(flag == 1) then {h_k := k;}
12.     else{h_k := h;}
13.     }
14.     g_k := e(c_k − b_k)/c_{h_k};
15.     foreach(j ∈ d(k)){
16.         set_g(T, α, eα_k ∏_{i∈d(k)\{j}} c_i, j, h_k, 0);
17.     }
18. }
```

Figure 1: Algorithms for determining $\widehat{\gamma}_{k,\alpha}(x^*)$, as returned from the procedure `main`.

Consider a parameterized family of distribution $\{\mathsf{P}_\phi : \phi \in \Phi\}$ with vector parameter ϕ, and let F be some function on Φ. We say that P_ϕ identifies $F(\phi)$ when $\mathsf{P}_\phi = \mathsf{P}_{\phi'}$ implies $F(\phi) = F(\phi')$. Here, F will be the identity, or some other projection of components of ϕ. In an MAR model, $\mathsf{P}_{\alpha,\theta}$ identifies (α, θ) iff

$$q_{\alpha,\theta}(x^*) = q_{\alpha',\theta'}(x^*), \forall x^* \in \Omega^* \Rightarrow (\alpha, \theta) = (\alpha', \theta'). \tag{44}$$

A simple example of MCAR data that is not identifiable is a two leaf tree in which, for each probe independently, data is missing from exactly one leaf. Then the only non-trivial equations (17) become

$$\begin{aligned} q_{\alpha,\theta}(\mathsf{1u}) &= \alpha_1\alpha_2\theta(\mathsf{1u}), \qquad & q_{\alpha,\theta}(\mathsf{0u}) &= (1-\alpha_1\alpha_2)\theta(\mathsf{0u}) \\ q_{\alpha,\theta}(\mathsf{u1}) &= \alpha_1\alpha_3\theta(\mathsf{u1}), \qquad & q_{\alpha,\theta}(\mathsf{u0}) &= (1-\alpha_1\alpha_3)\theta(\mathsf{u0}) \end{aligned} \tag{45}$$

The RHS of these equations are invariant w.r.t. the transformations $\alpha_1 \mapsto k\alpha_1$, $\alpha_2 \mapsto \alpha_2/k$, $\alpha_3 \mapsto \alpha_3/k$.

With each $S \subset R$ we associate the minimal logical multicast tree $\mathcal{T}_S = (V_S, L_S)$ that spans the root ρ and S. This is obtained by first finding the minimum spanning tree of ρ and S in $\mathcal{T}$. The branch points in the spanning tree, together with ρ and S, form the node set V_S. To define L_S, the parent $f_S(k)$ in $\mathcal{T}_S$, of each node in $U_S := V_S \setminus \{\rho\}$, is the $\succ$-minimal j in V_S such that $j \succ k$ in $\mathcal{T}$. A path in $\mathcal{T}$ that connects two nodes in V_S is called an S-segment. $K_S(i) = \{j \in V : i \preceq j \prec f_S(i)\}$ is the S-segment terminating at $i \in U_S$. Given $i \in V$, $\kappa_S(i)$ denote the node in V_S that terminates the S-segment containing i, i.e,. that for which $i \in K_S(\kappa_S(i))$. Likewise, $\alpha_S(i) = \prod_{j \in K_S(i)} \alpha_j$ denotes the composite transmission probability along the segment $K_S(i)$. $\alpha_S = \{\alpha_S(i) : i \in U_S\}$ will denote the collection of such probabilities.

Let D_S be the $\#U_S \times \#U$ incidence matrix of the nodes of U in the segments of $\mathcal{T}_S$, i.e., $D_{S,jk} = 1$ if $k \in K_S(j)$ and 0 otherwise. Setting $\beta_S(k) = \log \alpha_S(k)$ and $x_k = \log \alpha_k$ we have that

$$\beta_S = D_S x \tag{46}$$

for any $S \subset R$. Before stating and proving results on identifiability, we note that there exists at least one solution, $\log \alpha$, to (46). Let $\mathsf{P}_{\alpha,\theta,S}$ denote the distribution of the reports from nodes in S. We give two conditions for identifiability of α.

Theorem 7 *Let $\mathcal{T}$ be a canonical loss tree and $\{S_i\}_{i=1}^m$ a collection of subsets of R.*

(i) $U = \cup_{i=1}^n U_{S_i}$ if and only if the equations $\{\beta_{S_i} = D_{S_i} x\}_{i=1}^m$ have a unique solution x.

(ii) Assume $\mathsf{P}_{\alpha,\theta,S_i}$ identifies α_{S_i} for each i. Then $\{\mathsf{P}_{\alpha,\theta,S_i}\}_{i=1}^m$ identifies α iff either (and hence both) of the conditions of part (i) are satisfied.

Remarks. Uniqueness of the solution to (46) is determined by the structure of the D_S, which depend only on the topology and the choice of the S, not on β_S. Consequently, when uniqueness holds, it does so for any additive metric. Thus one can devise a test for identifiability based on path length in terms of number of links. Furthermore, if α is not identifiable, the procedure can be modified to determine which links can be solved.

We say that *complete data is available* from a subset S if $\theta(x^*) > 0$ for all x^* such that $R(\rho, x^*) = S$, i.e,. for which reports are received from all receivers in S and no others. Let $\mathcal{S}_\theta$ denote the set of subsets

S of R for which complete data is available, and Θ_c denote the set of missingness parameters θ for which $U = \cup_{S \in \mathcal{S}} U_S$.

Theorem 8 *Restrict the parameter space to $\mathcal{A} \times \Theta_c$ and assume data is MCAR. Then $\mathsf{P}_{\alpha,\theta}$ identifies (α, θ).*

Although we do not have a corresponding result for general MAR models, Theorem 8 is sufficient to enable further analysis of simple models in the following sections.

6 Asymptotics for Large Numbers of Probes

Let $\check{\alpha} = \arg\max_\alpha \mathcal{L}(\alpha)$ denote the incomplete data MLE arising from (20). In this section we examine the asymptotic properties of $\check{\alpha}$ as the number of probes n grows, without specific reference to the EM algorithm.

Theorem 9 *Assume data is MCAR. The incomplete data MLE $\check{\alpha}$ is consistent, i.e., $\lim_{n\to\infty} \check{\alpha} = \alpha$ almost surely.*

We now describe the asymptotic variance of $\check{\alpha}$ for large numbers of probes n in the regime of small loss probabilities $\overline{\alpha}$. We calculate the expected Fisher information matrix for the incomplete data problem, i.e., the matrix $\mathcal{I}(\alpha, \theta) = [\mathcal{I}^{ij}(\alpha, \theta)]_{ij \in U}$, where $\mathcal{I}^{ij}(\alpha) = -\mathsf{E}\frac{\partial^2 \mathcal{L}(\alpha)}{\partial \alpha_i \partial \alpha_j}$. Under conditions that we establish below, the inverse of $\mathcal{I}(\alpha)$, suitably rescaled, is the asymptotic variance of $\check{\alpha}$.

Our approach is to decompose the Fisher information matrix as a sum over subtrees for which complete data is present at the leaves. In the original incomplete data problem for the logical multicast topology $\mathcal{T}$, the counts $\boldsymbol{n}_S = \{m(x^*) \mid R(\rho, x^*) = S\}$, for each $S \subset R$, can be considered as a set of counts of complete outcomes on $\mathcal{T}_S$ stemming from those probes for which reports were received only from nodes in S. Thus the incomplete data log-likelihood function can then be decomposed as follows:

$$\mathcal{L}(\alpha) = \sum_{S \subset R: S \neq \emptyset} \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S), \quad \text{where} \quad \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S) = \sum_{x^*: R(\rho, x^*) = S} m(x^*) \log p_{\alpha_S}(x^*_S) \tag{47}$$

and $x^*_S = \{x^*_k : k \in S\}$ is the data in x^* that is observable at S. The corresponding decomposition of the expected Fisher information matrix is

$$\mathcal{I}^{ij}(\alpha) = \sum_{S \subset R: S \neq \emptyset} \sum_{k\ell \in U_S} \mathcal{I}^{k\ell}_S(\alpha_S) \frac{\partial \alpha_S(k)}{\partial \alpha_i} \frac{\partial \alpha_S(\ell)}{\partial \alpha_j} \tag{48}$$

where $\mathcal{I}^{jk}_S(\alpha_S) = -\mathsf{E}\frac{\partial^2 \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S)}{\partial \alpha_S(j) \partial \alpha_S(k)}$. Let $N_S = n\mathsf{P}[R(T^{(i)}(X^{(i)})) = S]$ be the mean number of probes with data observable exactly at S, and $W_S(i) = \sum_{j \in K_S(\kappa_S(i))} \overline{\alpha}_j$ the sum of link loss rates on the S-segment containing i.

Theorem 10 *(i) When $\theta \in \Theta_c$ and hence when $\check{\alpha}$ is consistent, $\sqrt{n}(\check{\alpha} - \alpha)$ converges in distribution to a mean zero multivariate Gaussian random variable with covariance matrix $n^{-1}\mathcal{I}(\alpha)$.*

(ii) When data is MCAR, $\mathcal{I}^{ij} = \sum_{S \subset R: S \neq \emptyset} \frac{N_S}{W_S(i)} \delta_{\kappa_S(i)\kappa_S(j)} + O(1)$ as $\overline{\alpha} \to 0$.

Example: uniform report transmission. Let reports be transmitted independently with uniform probability $p \in (0,1]$. Then $N_S = np^{\#S}\overline{p}^{\#R-\#S}$. For each $S \subset R$, and node $\ell \in U_S$, let $C_S(\ell)$ denote the matrix on U with entries $C_S^{ij}(\ell) = 1/W_S(i)$ if $i,j \in K_S(\ell)$ and 0 otherwise. For $s \in \{1,\ldots,\#R\}$ let $C_s = \sum_{S:\#S=s}\sum_{\ell\in U_S} C_S(\ell)$. Then

$$\mathcal{I} = nC\cdot(1+O(\overline{\alpha})), \quad \text{where} \quad C = \sum_{s=1}^{\#R} p^s(1-p)^{\#R-s}C_s \tag{49}$$

Let P_K denote the orthogonal projection onto the nullspace of a symmetric matrix K, and recursively define matrices $K_1,\ldots,K_{\#R}$ by $K_1 = C_1$, and

$$K_s = P_{K_1+\ldots K_{s-1}}C_sP_{K_1+\ldots K_s}, \quad K_1 = C_1. \tag{50}$$

Let r_0 denote the minimal s for which $P_{K_1+\ldots+K_s} = 1$. Since $C_{\#R} = 1$, such a $r_0 \le \#R$ exists.

Proposition 1 *$p^{r_0}C^{-1}$ converges to the pseudo-inverse of K_{r_0} as $p \to 0$.*

Let I_2 denote the Fisher information arising from measurements on binary subsets, i.e., I_2 is the sum obtained by restricting (48) to binary subsets S.

Proposition 2 *(i) $\mathcal{I} \ge \widetilde{\mathcal{I}}_2 > 0$, and hence $0 < \mathcal{I}^{-1} \le \mathcal{I}_2^{-1}$, in the order of positive linear operators.*

(ii) Proposition 1 holds with $r_0 = 2$.

Thus we have established:

Proposition 3 *Assume independent report loss with uniform probability p. Then $\sqrt{n}(\check{\alpha}-\alpha)$ converges to a multivariate Gaussian random variable with mean zero and covariance $G(\alpha,p)$, where $\lim_{p\to 0} p^2G(\alpha,p) = KI_2 + O(\|\overline{\alpha}\|^2)$ as $\overline{\alpha}\to 0$, where KI_2 is the pseudo-inverse of K_2.*

Remark: Proposition 3 suggests that we approximate the variance of $\check{\alpha}_k$ by $(KI_2)_{kk}/(np^2)$ when p and $\overline{\alpha}$ are small, and n is large.

Example: uniform report transmission from binary trees Consider the family of binary trees with 2^r leaves, $r = 1,2,\ldots$, with small uniform link loss probabilities $\overline{\alpha}$ and uniform small report transmission rate p. Let $v(r) = (v_1(r),\ldots,v_{r+1}(r))$ denote the set of unique diagonal elements of $KI_2/\overline{\alpha}$, the j^{th} element determining the asymptotic variance on links j nodes away from the root. Using Mathematica [16] to perform the algebra, we found the first six $v(r)$ to be:

$$\begin{aligned} v(1) &= \{\tfrac{1}{3},\tfrac{1}{3}\}, \quad v(2) = \{\tfrac{1}{8},\tfrac{21}{40},\tfrac{2}{5}\}, \quad v(3) = \{\tfrac{3}{80},\tfrac{49}{240},\tfrac{25}{42},\tfrac{3}{7}\}, \quad v(4) = \{\tfrac{1}{96},\tfrac{43}{672},\tfrac{27}{112},\tfrac{91}{144},\tfrac{4}{9}\}, \\ v(5) &= \{\tfrac{5}{1792},\tfrac{33}{1792},\tfrac{5}{64},\tfrac{21}{80},\tfrac{36}{55},\tfrac{5}{11}\}, \quad v(6) = \{\tfrac{3}{4096},\tfrac{187}{36864},\tfrac{133}{5760},\tfrac{153}{1760},\tfrac{73}{264},\tfrac{209}{312},\tfrac{6}{13}\} \end{aligned} \tag{51}$$

In all cases the estimator variance rises in a given tree on moving away from the root, except falling slightly at a leaf link as compared with its parent. At a given distance from the root, the link variance decreases as the tree depth increases. Both this trends can be understood by the intuition that variance should decrease

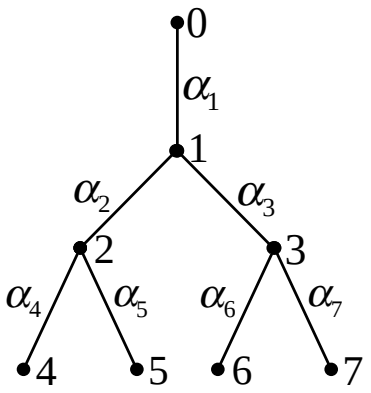


Figure 2: 4-RECEIVER BINARY TREE: used in model based simulation of Section 8.1.

when data is available from larger subtrees below a given link of interest. Considering the root and leaf links only, the values in (51) are consistent with the forms

$$v_1(r) = \frac{r}{r+2}\frac{1}{2^{2(r-1)}}, \qquad v_{r+1}(r) = \frac{r}{2r+1} \tag{52}$$

7 Convergence Rates for the EM iterates

We now consider convergence of the EM iterates themselves. Let $\mathcal{M}$ denote the map on $\mathbb{R}_+^U$ that implements the iteration, i.e., such that $\widehat{\alpha}^{(\ell+1)} = \mathcal{M}(\widehat{\alpha}^{(\ell)})$. A Taylor expansion of the iterative map gives

$$\widehat{\alpha}^{(\ell+1)} - \alpha \approx \nabla\mathcal{M} \cdot (\widehat{\alpha}^{(\ell)} - \alpha) \tag{53}$$

where $\nabla M_{ij} = \frac{\partial \mathcal{M}_i}{\partial \alpha_j}$ is the gradient of $\mathcal{M}$. A standard result [17, §3.9.3] expresses $\nabla\mathcal{M} = (1 - \mathcal{I}_c^{-1}\mathcal{I})$, with $\mathcal{I}_c$ the complete data information matrix and $\mathcal{I}$ the incomplete data information matrix from (48). The convergence ratio of the iteration is taken as the maximum eigenvalue λ for $\nabla\mathcal{M}$.

Our analysis of the convergence ratio is confined to the regime treated in Section 6, namely that of independent report transmission with small probability p, and small link loss probabilities $\overline{\alpha}$. In this regime, we have seen that $(\mathcal{I}_c^{-1})_{ij} = n^{-1}(\overline{\alpha}_i\delta_{ij} + O(\|\overline{\alpha}\|^2)$ and so from (49) $\nabla\mathcal{M}(\alpha)_{ij} = \delta_{ij} - \overline{\alpha}_i C_{ij} + O(\|\overline{\alpha}\|^2)$. Let $\mathcal{E}(X)$ denote the set of eigenvalues of a matrix X.

Proposition 4 *Assume independent report loss with probability $p \in (0,1)$ and small uniform probe loss rate $\overline{\alpha}$. The convergence rate λ for the EM algorithm obeys $\lambda = 1 - p^2\kappa + O(p^2(p+\overline{\alpha}))$. where κ is the minimum non-zero eigenvalue of $\overline{\alpha}K_2$.*

8 Experiments and Simulations

In order to evaluate the performance of the missing data inference algorithm, we conducted two types of simulation. First, we used model-based simulation in which the model for missing data indicators conformed

with the MCAR property. Second, we used a network-based implementation of the RTCP-based reporting mechanism outlined in Section 2.5. In this case the missing data indicators are not known to conform to the MAR model. This enabled us to test robustness of the algorithm with respect to violations of the MAR hypothesis that might occur in a real network application.

8.1 Model-based simulation

We conducted model based simulations on a balanced binary tree with 4 receivers, illustrated in Figure 2. Probe losses were independent with a uniform loss rate per link. Receiver reports were generated at each receiver for each probe and were transmitted independently with uniform probability p. We conducted 100 separate simulation runs, each of 100,000 probes. Initialization of $\widehat{\alpha}^{(0)}$ used (29). The termination criterion for the EM algorithm was that successive iterates $\widehat{\alpha}_k^{(\ell)}$ should have an absolute difference of less than 10^{-4} on each link k.

Figure 3(left) shows the mean and error bars for 95% confidence of link loss rate estimates obtained using the missing data algorithm. We also display the corresponding quantities for the complete data estimator applied to only those probes for which complete reports were available. In both cases the mean estimate is close to the model loss rate, i.e. $\overline{\alpha}_k = 0.01$. But note the rapid widening of error bars for the full data algorithm, compared with the missing data algorithm, as the report transmission probability decreases. From Prop. 3 we expect the standard error of the link loss rate estimates to diverge as p^{-1} for the missing data algorithm, regardless of the topology. However, in a 4-leaf tree the number of probes with complete data is proportional to p^4. Hence we expect the standard error to diverge as p^{-2}, with faster divergence for trees with more receivers. In this example, for p less than 0.4, the error bars encompass loss rate 0: the inferred loss from complete becomes statistically indistinguishable from 0.

Figure 3(right) breaks down the standard error of the link loss rate estimates according to the location of the link in the topology links 1, 2 and 4 being representative of links respectively 0, 1 and 2 links removed from the root. The experimental standard errors show close agreement with the theoretical values obtained by inverting the information matrix $\mathcal{I}$ in (49). We also show the small p approximation obtained using Proposition 3 and the values $v(2)$ from (51). The approximation remains reasonable even for quite large p.

8.2 RTCP-based experiments

The RTCP-based simulations used data gathered from a network-based implementation of loss reporting. Loss reports are embedded in RTCP feedback packets; any collector listening to these can then perform inference. The basic RTCP reporting mechanism includes only the average loss rate based on sequence numbers of received packets. An extension of the report format allows the inclusion of a binary vector indicating receipt or otherwise of a set of packets.

According to the RTP standard [20], the total report volume over all receivers should not exceed 5% of the source rate. RTCP clients estimate their share of this based upon the reports they hear from the other receivers, and limit report frequency and size accordingly. Consequently, for a sufficiently large number of receivers, it will not be possible to report on all probes. Missingness arises then by two mechanisms: the omission of certain probes from reporting, and the loss of report packets during transmission to the collector.

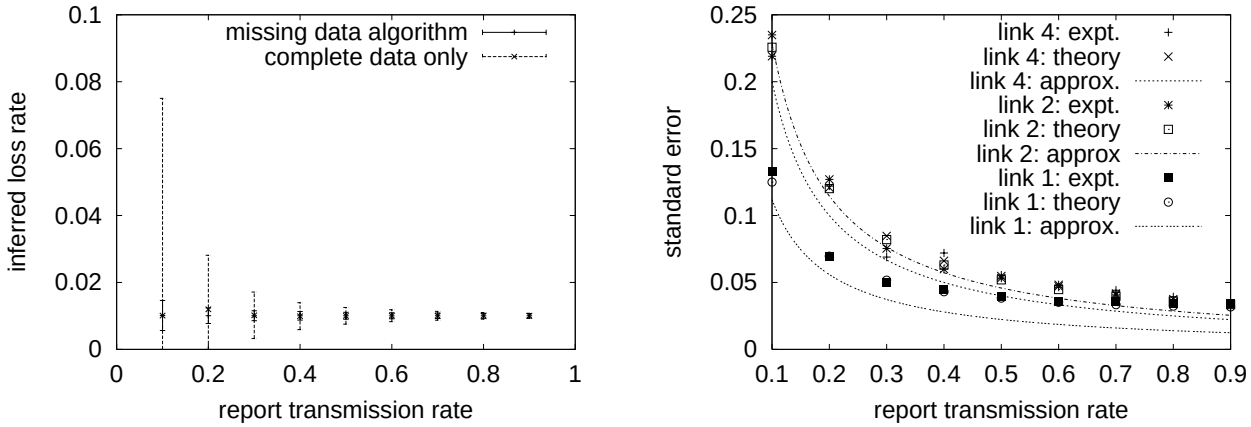


Figure 3: VARIANCE OF MISSING DATA ESTIMATOR IN 4-RECEIVER UNIFORM PROBE AND PACKET LOSS MODEL: Over 100 simulation runs each of 100,000 probes, uniform link loss $\overline{\alpha} = 0.01$, probe transmission rate p from 0.1 to 0.9. LEFT: mean estimate with error bars for 95% confidence. Comparison with estimator using only probes with complete data. RIGHT: standard error depending on link location: experiment, theory and approximation.

The implementation of extended RTCP-based reporting used in this study has a simulation mode that enables it to report on packet losses generated on a model topology according to a Bernoulli loss model, rather than due to packet loss in a real network. The probe source was chosen to have the characteristics of a GSM audio stream that could act as a probe source in real networks, sending packets at a rate of 50 per second. Since probe losses follow the assumed statistical model, only the missing data indicators can potentially exhibit departures from our model assumptions. Report thinning and transmission then takes place in the manner described above.

We collected traces from a 32 receiver balanced binary tree for which the link loss rates were chosen independently with a uniform distribution between 1% and 10%. The trace comprised reports on 11,956 probe packets, encompassing about 4 minutes at 50 packets per second. The mean number of reports received for a given probe was 18.8, so that the proportion of missing reports was 1 - 18.8/32 = 0.413. The maximum number of reports per probe was 29, i.e. no probe had complete data. Figure 4(left) shows a scatter plot of the 63 pairs of (actual,inferred) loss rates. The agreement is quiet close, with tight clustering around the line of slope 1 through the origin. The median relative error over all links was only 4.5%.

Figure 4(right) displays the median, 5^{th} and 95^{th} percentile of the relative error over all links as a function of the size of a subset of probes used for inference. Note that even with 2000 probes the relative error is typically less than 50%. Hence we can expect to identify the lossiest links with measurements over a duration less than 1 minute. The median error is only about 13% for this number of probes.

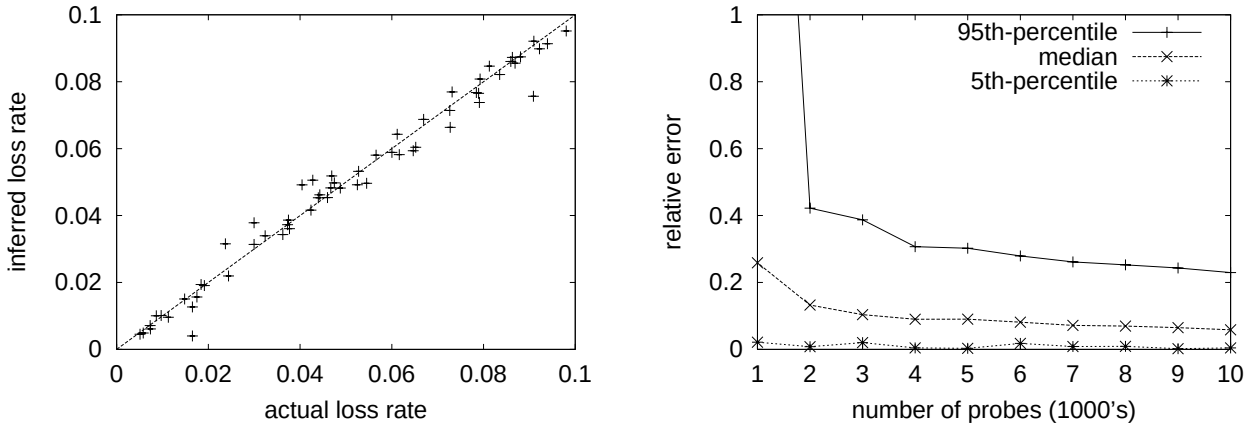


Figure 4: INFERENCE FORM RTCP TRACE DATA ON 32 RECEIVER BINARY TREE. LEFT: Scatter plot of inferred vs. actual loss rates for a full trace of 11,956 probes. RIGHT: Median, 5^{th} and 95^{th} percentiles of relative error over all links as function of number of probes.

9 Conclusions

In this paper we have extended the multicast based method for inferring network internal loss from end-to-end measurements that was first proposed in [2]. The original method assumed the presence of complete data specifying the set of end-points reached by each multicast probe. However, the proposed use of the RTCP transport protocol to transmit measurements inevitably leads to missing data, either through the need to thin data, or due to loss of reports in transmission. This motivated extending the former approach to work with missing data. An ad hoc approach of working with subsets of complete data would have several drawbacks: inference on all links may not be possible; inference would be inconsistent under the types of correlation between probe data and missingness that could reasonably occur in this context (see Section 2.5); and the estimators are not generally efficient. These considerations motivated the use of a more generally applicable scheme that accommodates the missing data directly, under more general conditions on the missing data mechanism.

This paper extended the Maximum-Likelihood approach of [2] to encompass missing data. We applied the EM algorithm to generate an iterative approximation to the corresponding MLE. We analyzed convergence rates for the EM algorithm itself, and for the MLE as the number of probes grows, and showed how to calculate these rates explicitly for a particular class of models. We tested an implementation of the algorithm in model-based simulations with known missingness statistics, and also in traces gathered from an implementation of the RTCP-based report transfer method. These results showed (i) the reduction in estimator variance, as compared with the ad hoc approach, where applicable; and (ii) accuracy of inferred loss rates compared with model or directly measured rates in the simulation; (iii) robustness of the approach under potential departures from the model assumptions on the missingness statistics in the RTCP-based application. In the RTCP-based experiments the median estimation error on a 32 receiver tree was only about 13% for 2,000 probes, and was typically less than about 50%. Thus inference sufficiently accurate to identify the

lossiest links could be performed on measurements collected over about a minute.

Future work is planned in two directions. First, we want to apply the same general methodology to the estimation of other internal characteristics, such as delay, utilization and topology itself, by adapting the framework of the present paper to work on estimation of these quantities with complete data, as performed in [3, 9, 15] A second direction is to develop more specific models of the missing data mechanism that could be used in a parametric approach to estimation with missing data. Lastly, we intend to publish elsewhere details of the RTCP reporting mechanism that motivated this study.

Acknowledgment We thank Francesco Lo Presti for some useful suggestions.

10 Proofs of Theorems

Proof of Theorem 2: We first render $\mathcal{L}_c(\alpha)$ into the canonical form of a standard exponential family.

- Denote by $\mathbf{0}$ and $\mathbf{1}$ the elements x of Ω with x_k all 0 or all 1 respectively.
- For $x \in \Omega$, denote by $W'(x)$ those nodes $k \in U$ for which $x_j = 0$ for all $j \in R(k)$. Let $W(x)$ be the $\succeq$-maximal elements of $W'(x)$. Note that $W(\mathbf{1}) = \emptyset$.
- For each $k \in U$ and $i \in \{0, 1\}$ define $q_k(i) = \mathsf{P}_\alpha[X_j = i, \forall j \in R(k) | X_{f(k)} = 1]$.
- Define new parameters $\{\delta_k : k \in U\}$ by $\delta_k = \log(q_k(0)/q_k(1))$.
- Observe that
$$\frac{p_\alpha(x)}{p_\alpha(\mathbf{1})} = \prod_{k \in W(x)} \frac{q_k(0)}{q_k(1)} = \prod_{k \in W(x)} e^{\delta_k}. \tag{54}$$
We interpret the product over $\emptyset$ for $x = \mathbf{1}$ as 1.
- The map taking $\mathcal{A}$ to its image Δ under the change of parameters $\alpha \mapsto \delta$ is invertible. To see this note that given $\sum_{x \in \Omega} p_\alpha(x) = 1$, (54) fixes the $p_\alpha(x)$ in terms of the δ. These in turn determine the γ_k, and hence the α_k, by Theorem 2.
- Writing $n(1) = n - \sum_{x \neq \mathbf{1}} n(x)$, and recalling that $W(\mathbf{1}) = \emptyset$, we find
$$\mathcal{L}_c(\alpha) \;=\; \sum_{x \in \Omega} n(x) \sum_{k \in W(x)} \delta_k + n \log p_\alpha(\mathbf{1}) = \sum_{k \in U} N_k \delta_k - n c(\delta),$$
where $N_k = \sum_{x \in \Omega : k \in W(x)} n(x)$, and $c(\delta)$ is the reparameterization of $-\log p_\alpha(\mathbf{1})$ in terms of δ:
$$c(\delta) = \log \sum_{x \in \Omega} \prod_{k \in W(x)} e^{\delta_k}. \tag{55}$$

The expression (55) has the form of a standard exponential family, with the log-likelihood expressed in terms of the natural parameters $\delta = \{\delta_k : k \in U\}$, sufficient statistics $N = \{N_k : k \in U\}$, and cumulant

$c(\delta)$. Since $c(\delta)$ is finite for all $\delta \in \mathbb{R}^U$ the family can be considered full. However, the parameter space of interest is the open subset $\Delta \subset \mathbb{R}^U$ that is the image of $\mathcal{A}$ under the reparameterization $\alpha \mapsto \delta$.

Since the mapping $\mathcal{A} \to \Delta : \ \alpha \mapsto \delta$ is invertible, the parameters δ are identifiable by Theorem 2(v), and hence the exponential family is affinely independent. (A simple argument shows that natural parameters in an open set are identifiable iff the exponential family is affinely independent). A well-known result (see e.g. [13, Ex. 6.6.3.]) for standard exponential families then says that the MLE is the solution δ of

$$N_k = \mathsf{E}_{\delta'}[N_k], \quad k \in U \tag{56}$$

provided this δ lies in the interior of Δ. But clearly $\sum_{j \succeq k} N_j = n(1 - \widehat{\gamma}_k)$, and hence finding the solution to (56) is equivalent to finding the solution α' to

$$\gamma_k = \mathsf{E}_{\alpha'}[\widehat{\gamma}_k], \quad k \in U. \tag{57}$$

Provided $\widehat{\gamma} \in \mathcal{G}$, then by Theorem 1, $\widehat{\alpha}$ is the unique such solution, and hence if it lies in $\mathcal{A}$ it is the MLE.∎

Proof of Theorem 3: Observe that $\mathsf{E}_\alpha[\mathcal{L}_c(\alpha')|\boldsymbol{m}] = \sum_{x \in \Omega} \mathsf{E}_\alpha[n(x)|\boldsymbol{m}] \log p_{\alpha'}(x)$. Hence maximizing $Q_{\widehat{\alpha}^{(\ell)},\alpha'}$ over α' is equivalent to finding the complete-data MLE, but with $n(x)$ replaced by $\mathsf{E}_{\widehat{\alpha}^{(\ell)}}[n(x)|\boldsymbol{m}]$ throughout. In particular, $\widehat{\gamma}_k$, being a linear combination of the $n(x)$, gets replaced by $\widehat{\gamma}_k^{(\ell)}$. Now if $\widehat{\alpha}^{(\ell)} \in \mathcal{A}$ then it is not hard to see that $\widehat{\gamma}_{\widehat{\alpha}^{(\ell)}}(x^*) \in \mathcal{G}$ for each x^*, and hence $\widehat{\gamma}^{(\ell)} \in \mathcal{G}$ since $\mathcal{G}$ is convex. The result then follows from Theorems 1 and 2. ∎

Proof of Theorem 4: The condition says that the sequence of EM iterates exists in $\mathcal{A}$. Parts (i) and (ii) follow from Theorem 6 of [22]. This is because: (a) by Theorem 1(iii), $\widehat{\alpha}^{(\ell+1)}$ is stationary for $\alpha \mapsto Q(\alpha, \widehat{\alpha}^{(\ell)})$; (b) $\nabla_\alpha Q(\alpha', \alpha)$ is clearly continuous for $(\alpha', \alpha) \in \mathcal{A} \times \mathcal{A}$; and (c) By Theorem 2, $\mathcal{L}_c$ comes from a regular exponential family and hence $\|\widehat{\alpha}^{(\ell+1)} - \widehat{\alpha}^{(\ell)}\| \to 0$ as $\ell \to \infty$; see remark 3(vi) in [22]. Part (iii) then follows from Corollary 1 in [22]∎

Proof of Theorem 5: For the proof we observe the following Markov property of the probe process X:

(M) Conditioned on $X_i = 1$, the distributions of the sets of variables $\{X_k : k \in R(i)\}$ are independent for different $i \in d(k)$.

If $y_k^* = 1$, then $\vee_{j \in R(k)} x_j = 1$, and hence $\widehat{\gamma}_{k,\alpha}(x^*) = 1$. Suppose instead that $y_k^* < 1$. Let $Q(k)$ denote the event that $X_j = 0$ for all $j \in R(k, x^*)$ if $R(k, x^*) \neq \emptyset$, otherwise take $Q(k)$ as the universal set in the underling probability space. Since $y_h^* = 1$ for $h = h(k, x^*)$, then $\{X^* = x^*\} \subset \{X_h = 1\}$ and hence $\{X^* = x^*\} = \{X_h = 1\} \cap Q(d(h,k)) \cap \{X_j^* = x_j^* : j \in R \setminus R(d(h,k))\}$. This yields that

$$\mathsf{P}_\alpha[\vee_{j \in R(k)} X_j = 1 \mid X^* = x^*] = \frac{\mathsf{P}_\alpha[\{\vee_{j \in R(k)} X_j = 1\} \cap Q(d(k,h)) \mid X_h = 1]}{\mathsf{P}_\alpha[Q(d(k,h)) \mid X_h = 1]}, \tag{58}$$

where we have used property (M), and the fact that $\mathsf{P}[A \mid B \cap C \cap D] = \mathsf{P}[A \mid C \mid B]$ when A and C are conditionally independent of D given B.

The denominator in (58) is just $c_{d(h,k)}$. To treat the numerator, observe that

$$\begin{aligned}
&\mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(i) \mid X_{f(i)} = 1] \qquad (59)\\
&\quad = \mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(d(k,i)) \cap \{X_i = 1\} \cap_{j\in d(i)\setminus d(k,i)} Q(j) \mid X_{f(i)} = 1]\\
&\quad = \mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(d(k,i)) \mid X_i = 1 \mid X_{f(i)} = 1]\mathsf{P}_\alpha[X_i = 1 \mid X_{f(i)} = 1] \prod_{j\in d(i)\setminus d(k,i)} \mathsf{P}_\alpha[Q(j) \mid X_i \mid X_{f(i)}]\\
&\quad = \mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(d(k,i)) \mid X_i = 1] \ \alpha_i \prod_{j\in d(i)\setminus d(k,i)} c_j.
\end{aligned}$$

Here we have used $Q(j) = \cap_{i\in d(j)} Q(i)$, the Markov property (M), and the fact that $\{X_k = 1\} \subset \{X_{f(k)} = 1\}$. Applying (59) repeatedly to the numerator of (58), we obtain the form $\mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(k) \mid X_{f(k)} = 1] \prod_{k\prec i\preceq d(k,h)} \alpha_i \prod_{j\in d(i)\setminus\{d(i,k)\}} c_j$.

The first term is in this product is $\mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \cap Q(k) \mid X_{f(k)} = 1] = c_k \mathsf{P}_\alpha[\{\vee_{j\in R(k)} X_j = 1\} \mid Q(k) \mid X_{f(k)} = 1] = c_k(1 - \mathsf{P}_\alpha[\vee_{j\in R(k)} X_j = 0 \mid X_{f(k)} = 1]/c_k) = c_k - b_k$, and the stated result (35) follows. ■

Proof of Lemma 1: We first show that for each $k \in V$ there is some $x^* \in \Omega_0^*$ with $m(x^*) > 0$ for which $\widehat{\gamma}_{k,\alpha}(x^*) > 0$. By condition (38) there is x^* with $m(x^*) > 0$, for which either $x_j^* = 1$ or $x_j^* = \mathsf{u}$ for some $j \in R(k)$. In the former case $\widehat{\gamma}_{k,\alpha}(x^*) = 1-$ or $x_j^* = \mathsf{u}$; in the latter case it is not hard to see in Theorem 5 that $c_k > b_k$ and hence $\widehat{\gamma}_{k,\alpha}(x^*) > 0$. Since $\widehat{\gamma}_{k,\alpha}(x^*) \geq 0$ for all $x^* \in \Omega_0^*$, then $\widehat{\gamma}_{k,\alpha} > 0$.

By (40) then for each $k \in V \setminus R$ there exists x^* with $m(x^*) > 0$ such that $\widehat{\gamma}_{k,\alpha}(x^*) = 1$, and children j, ℓ of k for which $\widehat{\gamma}_{j,\alpha}(x^*) = 1$ while $\widehat{\gamma}_{j,\alpha}(x^*) > 0$. Hence $\widehat{\gamma}_{k,\alpha}(x^*) < \sum_{j\in d(k)} \widehat{\gamma}_{j,\alpha}(x^*)$. Since by definition $\widehat{\gamma}_{k,\alpha}(x^*) \leq \sum_{j\in d(k)} \widehat{\gamma}_{j,\alpha}(x^*)$ for all x^*, we have $\widehat{\gamma}_{k,\alpha} < \sum_{j\in d(k)} \widehat{\gamma}_{j,\alpha}$.

Taking these relations over all relevant k, we conclude that $\widehat{\gamma}_\alpha \in \mathcal{G}$. ■

Proof of Theorem 6: Reparameterizing the incomplete data likelihood function $\mathcal{L}$ in terms of γ, we obtain

$$\begin{aligned}
\mathcal{L} \quad = \quad & n(11)\log(\gamma_2 + \gamma_3 - \gamma_1) + n(10)\log(\gamma_1 - \gamma_3) + n(01)\log(\gamma_1 - \gamma_2) + n(00)\log(1 - \gamma_1)\\
& + n(1\mathsf{u})\log(\gamma_2) + n(\mathsf{u}1)\log(\gamma_3) + n(0\mathsf{u})\log(1 - \gamma_2) + n(\mathsf{u}0)\log(1 - \gamma_3)
\end{aligned}$$

Writing this form as $\mathcal{L} = \sum_{x*\in\Omega_0^*} n(x^*)\mathcal{L}_{x^*}$, it suffices to show that $-\mathcal{L}_{x^*}$ is jointly convex in $\gamma_1, \gamma_2, \gamma_3$ for each x^*. This follows from the fact, established by direct computation, that the principal minors of the Hessian matrices $-[\partial^2 \mathcal{L}_{x^*}/\partial\gamma_i\partial\gamma_j]_{ij=1,2,3}$ are non-negative. Convergence then follows from Theorem 4 and the standing assumptions (38) and (40). ■

Remark: the method used in the proof appears not to extend to more general trees, not even binary ones.

Proof of Theorem 7: (i) We establish that if $U = \cup_{i=1}^m U_{S_i}$, then there is a unique solution x. We do this by contradiction. Suppose that there exists a node $k \in V$ such that there does not exist a unique

value for x_k. Pick any such $\prec$-maximal k. By assumption, k terminates some segment $K_{S_i}(k)$ and hence $x_k = \beta_{S_i}(k) - \sum_{k \prec j \prec f_{S_i}(k)} x_j$. By the maximality assumption, all terms on the RHS are unique, and hence so is x_k.

We establish now that if there is a unique solution x, then $U = \cup_{i=1}^m U_{S_i}$. This is done by contradiction. First, note if the solution x is unique, it must be $x_k = \log \alpha_k$. Second, note that we only need to consider a branch point $k \in U \setminus R$. Assume there exists $k \notin \cup_{i=1}^m U_{S_i} \setminus R$. Then every segment containing $v \in d(k)$ also contains k. Using $x_k = \log \alpha_k$ we can reduce the family of equations $\{\beta_{S_i} = D_{S_i} x\}_{i=1}^m$ to the following set of equations describing the behavior of x_k and x_v, $v \in d(k)$,

$$x_k + x_v = \beta'_v, \qquad \forall v \in d(k) \tag{60}$$

where β'_v is the log of the probability of a reception of a packet at $v \in d(k)$ given that it was received at $f(k)$. The β'_v are determined by the link probabilities other than α_k and α_v, $v \in d(k)$, and the original β_{S_i}. However, these equations do not have a unique solution, resulting in a contradiction.

(ii) The solution x to $\{\beta_{S_i} = D_{S_i} x\}_{i=1}^m$ is not unique if and only if there is more that one set of link probabilities α giving rise to the same $\{\alpha_{S_i}\}_{i=1}^m$ and hence the same $\{\mathsf{P}_{\alpha,\theta,S_i}\}_{i=1}^m$. ∎

Proof of Theorem 8: With MCAR the missingness probabilities do not depend on any data, and so we can replace $\theta(x^*)$ with $\theta(t(x^*))$. Since $\sum_{x^*:t(x^*)=t'} p^*_\alpha(x^*) = 1$ for all $t' \in \{0,1\}^R$, $q_{\alpha,\theta}(x^*) = q_{\alpha',\theta'}(x^*), \forall x^* \in \Omega^*$ implies $\theta = \theta'$.

Consider now any $S \subset R$ for which complete data is available. Then $\theta(t(x^*))$ are equal and hence strictly positive for any x^* with $R(\rho, x^*) = S$. Hence $q_{\alpha,\theta}(x^*) = q_{\alpha',\theta'}(x^*)$ implies $p_\alpha(x^*) = p_{\alpha'}(x^*)$. By Theorem 1(iv), α_S is identifiable. Then α is identifiable by Theorem 7(ii) since $\theta \in \Theta_c$. ∎

Proof of Theorem 9: The proof of mirrors that of Theorem 3(ii) in [2] which in turn uses Lemma 7.54 in [19]. Although not mentioned there, the latter result requires identifiability. This follows from the MCAR assumption and Theorem 8. ∎

Proof of Theorem 10: We first show that $\mathcal{I}_S$ is positive definite if complete data is available from S. By standard arguments (see e.g. Prop 2.84 in [19]) one writes $\mathcal{I}_S^{jk}(\alpha_S) = \mathsf{Cov}_{\alpha,\theta}(\frac{\partial \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S)}{\partial \alpha_S(j)}, \frac{\partial \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S)}{\partial \alpha_S(k)})$. If $\mathcal{I}_S^{jk}(\alpha_S)$ is not positive definite, there exists some nonzero $c \in \mathbb{R}^{U_S}$ for which $c \cdot \mathcal{I}_S(\alpha_S) \cdot c = \mathsf{Var}_{\alpha,\theta}(c \cdot \frac{\partial \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S)}{\partial \alpha_S}) = 0$. This happens if $c \cdot \frac{\partial \mathcal{L}_c(\mathcal{T}_S, \boldsymbol{n}_S, \alpha_S)}{\partial \alpha_S} = 0$, almost surely, or, equivalently, if $c \cdot \frac{\partial \log p_{\alpha_S}(x^*_S)}{\partial \alpha_S} = 0$ for all x^*_S, since $\theta(x^*) > 0$ by assumption. Repeating the argument of Theorem 4(ii) in [2], this requires $c = 0$.

Observe that $\frac{\partial \alpha_S}{\partial \alpha} = B(\alpha_S) D_S B^{-1}(\alpha)$ where $B(\alpha)$ is the diagonal matrix with entries from α. Thus $\mathcal{I} \geq \sum_{S \in \mathcal{S}_\theta} B^{-1}(\alpha) D_S^T B(\alpha_S) I_S(\alpha_S) B(\alpha_S) D_S B^{-1}(\alpha)$, in the order of positive linear operators. Since the kernel of a sum of non-negative definite operators is the intersection of their kernels, the sum is positive definite iff $\cap_{S \in \mathcal{S}_\theta} \ker(D_S) = \{0\}$, which happens iff the equations $\{\beta_S = D_S x\}_{S \in \mathcal{S}_\theta}$ have a unique solution x, which is guaranteed by Theorem 7(i) and the assumption that $\theta \in \Theta_c$.

(ii) Let $S \in \mathcal{S}_\theta$. Note $\alpha_S(k)/\alpha_i = D_{S,ki}(1+O(\|\overline{\alpha}\|))$. With data MCAR, the $\theta(x^*)$ are equal for any x^* with $R(\rho, x^*) = S$, and hence $\mathsf{E}m(x^*) = N_S p^*_\alpha(x^*) = N_S p_{\alpha_S}(x^*_S)$. From Theorem 5 of [2], $N_S^{-1}\mathcal{I}_S(\alpha_S)$ has inverse ν_S for which $\nu_S^{k\ell} = \overline{\alpha}_S(k)\delta_{k\ell} + O(\|\overline{\alpha}_S\|^2)$. Hence $\mathcal{I}_S^{k\ell}(\alpha_S) = \frac{N_S}{\overline{\alpha}_S(k)}(\delta_{k\ell} + O(\|\overline{\alpha}_S\|)$. Finally, observe that $\overline{\alpha}_S(\kappa_S(i)) = W_S(i) + O(\|\overline{\alpha}\|^2)$. Putting these together with (48) the result follows. ∎

Proof of Proposition 1: Rewrite $C = \sum_{s=1}^{\#R} p^s C'_s$ where $C'_s = \sum_{1\le t\le s} C_t \cdot (-1)^{s-t}\binom{n-t}{s-t}$. Observe that the definition of K_s is invariant under replacement of C_s by C'_s in (50) since $P_{K_1+\ldots+K_{s-1}}C_{s'}P_{K_1+\ldots+K_{s-1}} = 0$ for $s' < s$. Let W_s denote the unitary matrix that diagonalizes K_s and set $W = \prod_{i=1}^{r_0} W_s$. (Since the ranges of the K_s are disjoint, the various W_s commute). Then up to unitary transformation under W, we can write C is block diagonal form

$$C = W \begin{pmatrix} pA_{11}(p) & p^2A_{12}(p) & \cdots & p^{r_0}A_{1r_0}(p) \\ p^2A_{21}(p) & p^2A_{22}(p) & \cdots & p^{r_0}A_{2r_0}(p) \\ \vdots & \vdots & \ddots & \vdots \\ p^{r_0}A_{r_01}(p) & p^{r_0}A_{r_02}(p) & \cdots & p^{r_0}A_{r_0r_0}(p) \end{pmatrix} W^T \tag{61}$$

where $A_{ij}(p) = A_{ji}^T(p)$, and each submatrix A_{ij} converges to some A_{ij} as $p \to 0$ such that the block matrix with A_{ss} as the s^{th} diagonal element and zero elsewhere is unitarily equivalent to K_s under W. By construction, the A_{ss} are invertible, and hence so are the $A_{ss}(p)$ for sufficiently small p. The block diagonal representation of C can be inverted inductively as follows. Assume the block submatrix $B_{[s-1][s-1]}$ comprising the first $s-1$ blocks can be inverted and that $B^{-1}_{[s-1][s-1]}$ is $O(p^{1-s})$ as $p \to 0$. This condition is trivially satisfied for $s = 2$. Now write $B_{[s][s]}$ as a two-by-two superblock matrix

$$B_{[s][s]} = \begin{pmatrix} B_{[s-1][s-1]} & B_{[s-1]\,s} \\ B_{s\,[s-1]} & p^sA_{ss}(p) \end{pmatrix} \tag{62}$$

Then $B^{-1}_{[s][s]} = D_{s,1}(p) \cdot D^{-1}_{s,2}(p)$ where

$$D_{s,1}(p) = \begin{pmatrix} B^{-1}_{[s-1][s-1]} & -p^{-s}B^{-1}_{[s-1][s-1]}B_{[s-1]\,s}A^{-1}_{ss}(p) \\ -p^{-s}A^{-1}_{ss}(p)B_{s\,[s-1]}B^{-1}_{[s-1][s-1]} & p^{-s}A^{-1}_{ss}(p) \end{pmatrix} \tag{63}$$

$$D_{s,2}(p) = \begin{pmatrix} 1 - p^{-s}B_{[s-1]\,s}A^{-1}_{ss}(p)B_{s\,[s-1]}B^{-1}_{[s-1][s-1]} & 0 \\ 0 & 1 - p^{-s}B_{s\,[s-1]}B^{-1}_{[s-1][s-1]}B_{[s-1]\,s}A^{-1}_{ss}(p) \end{pmatrix} \tag{64}$$

Now $B_{[s-1],s} = O(p^s)$ as $p \to 0$, from which it follows that

$$B_{[s,s]}(p) = \begin{pmatrix} 0 & 0 \\ 0 & p^{-s}A^{-1}_{ss} \end{pmatrix} + O(p^{1-s}). \tag{65}$$

Consequently, $B^{-1}_{[ss]} = O(p^{-s})$ as $p \to 0$, completing the induction step. The statement of the Proposition then follows from (65) by taking $s = r_0$. ∎

Proof of Proposition 2: (i) In the given model, $N_S > 0$ for all subsets S of R, and hence $\mathcal{I} \ge \mathcal{I}_2$. Since each node in $U \subset R$ has at least two descendent leaves $U = \cup_{\#S=2}U_S$ and hence $\mathcal{I}_2 > 0$ by an argument

similar to that in Theorem 7(i). Now if $A \geq B > 0$ then $0 < 1 - a \leq 1 - b < 1$ where $a = A/(2\|A\|)$ and $b = B/(2\|A\|)$. Since $\|1-a\|,\ \|1-b\| < 1$, $a^{-1} = (1-(1-a))^{-1} = \sum_{i=0}^{\infty}(1-a)^i < \sum_{i=0}^{\infty}(1-b)^i = b^{-1}$, whence the result follows.

(ii) Consider inference performed by using only measurements from binary sets S. $\mathcal{I}_2 = pQ_1 + p^2Q_2$ for some Q_1 and Q_2 independent of p, and hence $\mathcal{I}_2^{-1}$ is $O(p^{-2})$. By (i), $\mathcal{I}^{-1} \leq \mathcal{I}_2^{-1}$, which precludes $r_0 > 2$ in Proposition 1. We conclude $r_0 = 2$ by showing that K_1 has 0 as an eigenvalue, for then $r_0 > 1$. Assume that the root ρ as a unique child 1. If not, partition the $\mathcal{T}$ into disjoint subtrees with nodes descended from each child of ρ, then apply the following argument to each subtree. Let v denote the element of R^U with $v(1) = 1$, $v(j) = -1$ for $j \in d(1)$, and $v(k) = 0$ otherwise. Observe that for each $k \in R$, the $C_{\{k\}}(k)_{ij}$ are equal for $i, j \in \{1, d(k,1)\}$. Since $C_1 = \sum_{k\in R} C_{\{k\}}$, then $C_1 \cdot v = 0$. ∎

Proof of Proposition 4: $\max \mathcal{E}(\nabla\mathcal{M}) = 1 - \overline{\alpha} \min \mathcal{E}(C) + O(\overline{\alpha}))$. From Propositions 1 and 2 we know that C takes the block diagonal form (61) with $r_0 = 2$. From this is follows that each eigenvalue of C takes the form $p^i v_i(p) + O(p^{i+1})$ for some $i \in \{1, 2\}$, where $\lim_{p\to 0} v_i(p) \in \mathcal{E}(A_{ii})$. Since $0 \notin \mathcal{E}(A_{ii})$, $p^{-2} \min \mathcal{E}(C) \to \min \mathcal{E}(A_{22})$ as $p \to 0$, and hence the result follows. ∎

References

[1] A. Adams, T. Bu, R. Cáceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior, IEEE Communications Magazine, May 2000.

[2] R. Cáceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics" IEEE Trans. on Information Theory, vol. 45, pp. 2462-2480, 1999.

[3] R. Cáceres, N.G. Duffield, J.Horowitz F. Lo Presti, D. Towsley, "Statistical Inference of Multicast Network Topology", in Proc. IEEE Conf. on Decision and Control, Phoenix, AZ, Dec. 1999.

[4] R. Carter, M. Crovella, "Measuring bottleneck link-speed in packet-switched networks," *Performance Evaluation*, 27&28, 1996.

[5] M. Coates, R. Nowak. "Network loss inference using unicast end-to-end measurement", *Proc. ITC Conf. IP Traffic, Modeling and Management*, Sept. 2000.

[6] M.J. Coates and R. Nowak, "Network Delay Distribution Inference from End-to-end Unicast Measurement," to appear in *Proc. IEEE ICASSP*, May 2001.

[7] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)", J. Roy. Statist. Soc. Ser., vol. 39, pp. 1–38, 1977.

[8] A.B. Downey. "Using pathchar to estimate Internet link characteristics," *Proc. SIGCOMM'99* Sept. 1999.

[9] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", in Proc. IEEE Infocom 2000, Tel Aviv, March 2000.

[10] N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley, "Inferring link loss using striped unicast probes," to appear in Proc. IEEE Infocom 2001, Anchorage, Alaska, April 22-26, 2001.

[11] T. Friedman, R. Cáceres, K. Almeroth, K. Sarac, "RTCP Reporting Extensions", work in progress, January 2001.

[12] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see `ftp://ftp.ee.lbl.gov/pathchar`

[13] E.L. Lehmann and G. Casella, "Theory of Point Estimation", Springer Texts in Statistics, Springer, New York, 1998.

[14] R.J.A. Little and D.B. Rubin, "Statistical Analysis with Missing Data", Wiley, New York, 1987

[15] F. Lo Presti, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[16] Mathematica. See www.wolfram.com

[17] G.J. McLachlan and T. Krishnan, "The EM algorithm and extensions", Wiley, New York, 1997.

[18] `ns` – Network Simulator. See http://www-mash.cs.berkeley.edu/ns/ns.html

[19] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[20] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, January 1996.

[21] Y. Shavitt, X. Sun, A. Wool, B. Yener. "Computing the Unmeasured: An Algebraic Approach to Internet Mapping" to appear in *Proc. INFOCOM01*.

[22] C.F. Jeff Wu, "On the convergence properties of the EM algorithm", Annals of Statistics, vol. 11, pp. 95–103, 1982

Multicast-Based Inference of Network-Internal Delay Distributions*

F. Lo Presti[†,§] N.G. Duffield[†] J. Horowitz[‡] D. Towsley[§]

[†]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
{lopresti,duffield}@research.att.com

[‡]Dept. Math. & Statistics
University of Massachusetts
Amherst, MA 01003, USA
joeh@math.umass.edu

[§]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@gaia.cs.umass.edu

September 15, 1999

Abstract

Packet delay greatly influences the overall performance of network applications. It is therefore important to identify causes and location of delay performance degradation within a network. Existing techniques, largely based on end-to-end delay measurements of unicast traffic, are well suited to monitor and characterize the behavior of particular end-to-end paths. Within these approaches, however, it is not clear how to apportion the variable component of end-to-end delay as queueing delay at each link along a path. Moreover, they suffer of scalability issues if a significant portion of a network is of interest.

In this paper, we show how end-to-end measurements of multicast traffic can be used to infer the packet delay distribution and utilization on each link of a logical multicast tree. The idea, recently introduced in [4, 5] is to exploit the inherent correlation between multicast observations to infer performance of paths between branch points in a tree spanning a multicast source and its receivers. The method does not depend on cooperation from intervening network elements; because of the bandwidth efficiency of multicast traffic, it is suitable for large scale measurements of both end-to-end and internal network dynamics. We establish desirable statistical properties of the estimator, namely consistency and asymptotic normality. We evaluate the estimator through simulation and observe that it is robust with respect to moderate violations of the underlying model.

Keywords. End-to-end measurements, queueing delay, estimation theory, multicast tree, network tomography.

*This work was sponsored in part by the DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

1 Introduction

Background and Motivation. Monitoring the performance of large communications networks is essential for diagnosing the causes of performance degradation. There are two broad approaches to monitoring. In the *internal* approach, direct measurements are made at or between network elements, e.g. of packet loss or delay. In the *external* approach, measurements are made across a network on end-to-end or edge-to-edge paths.

The internal approach has a number of potential limitations. Due to the commercial sensitivity of performance measurements, and the potential load incurred by the measurement process, it is expected that measurement access to network elements will be limited to service providers and, possibly, selected peers and users. The internal approach assumes sufficient coverage, i.e. that measurements can be performed at all relevant elements on paths of interest. In practice, not all elements may possess the required functionality, or it may be disabled at heavily utilized elements in order reduce CPU load. On the other hand, arranging for complete coverage of larger networks raises issues of scale, both in the in the gathering of measurement data, and joining data collected from a large number of elements in order to form a composite view of end-to-end performance.

This motivates external approaches, network diagnosis through end-to-end measurements, without necessarily assuming the cooperation of network elements on the path. There has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [3, 9, 19, 26, 27, 29]). Several research efforts are working on the developments of measurement infrastructure projects (Felix [13], IPMA [15], NIMI [18] and Surveyor [35]) with the aim to collect and analyze end-to-end measurements across a mesh of paths between a number of hosts. Standard diagnostic tools for IP networks, `ping` and `traceroute` report roundtrip loss and delay, the latter incrementally along the IP path by manipulating the time-to-live (TTL) field of probe packets. A recent refinement of this approach, `pathchar` [17], estimates hop-by-hop link capacities, packet delay and loss rates. `pathchar` is still under evaluation; initial experience indicates many packets are required for inference leading to either high load of measurement traffic or long measurement intervals, although adaptive approaches can reduce this [10]. More broadly, measurement approaches based on TTL expiry require the cooperation of network elements in returning Internet Control Message Protocol (ICMP) messages. Finally, the success of active measurement approaches to performance diagnosis may itself cause increased congestion if intensive probing techniques are widely adopted.

In response to some of these concerns, a multicast-based approach to active measurement has been proposed recently in [4, 5]. The idea behind the approach is that correlation in performance

seen on *intersecting* end-to-end paths can be used to draw inferences about the performance characteristics of the common portion (the intersection) of the paths, without the cooperation of network elements on the path. Multicast traffic is particular well suited for this since a given packet only occurs once on a given link in the (logical) multicast tree. Thus characteristics such as loss and end-to-end delay of a given multicast packet as seen at different endpoints are highly correlated. Another advantage of using multicast traffic is scalability. Suppose packets are exchanged on a mesh of paths between a collection of N measurement hosts stationed in a network. If the packets are unicast, then the load on the network may grow proportionally to N^2 in some parts of the network, depending on the topology. For multicast traffic the load grows proportionally only to N.

Contribution The work of [4, 5] showed how multicast end-to-to measurements can be used to infer per link loss rates in a logical multicast tree. In this paper we extend this approach to infer the probability distribution of the per link variable delay. Thus we are not concerned with propagation delay on a link, but rather the distribution of the additional variable delay that is attributable to either queuing in buffers or other processing in the router. A key part of the method is an analysis that relates the probabilities of certain events visible from end-to-end measurements (end-to-end delays) to the events of interest in the interior of the network (per-link delays). Once this relation is known, we can estimate the delay distribution on each link from the measured distributions of end-to-end delays of multicast packets.

For a glimpse of how the relations between end-to-end delay and per link delays could be found, consider a multicast tree spanning a source of multicast probes (identified as the root of the tree) and a set of receivers (one at each leaf of the tree). We assume the packets are potentially subject to queuing delay and even loss at each link. Focus on a particular node k in the interior of the tree. If, for a given packet, the source-to-leaf delay does not exceed a given value on any leaf descended from k, then clearly the delay from the root to the node k was less than that value. The stated desired relation between the distributions of per-link and source-to-leaf delays is obtained by a careful enumeration of the different ways in which end-to-end delay can be split between the portion of the path above or below the node in question, together with the assumption that per-link delays are *independent* between different links and packets. We shall comment later upon the robustness of our method to violation of this independence assumption.

We model link delay by non-parametric discrete distributions. The choice of non parametric distributions rather than a parameterized delay model is dictated by the lack of knowledge of the distribution of link delays in networks. While there is significant prior work on the analysis and characterization of end-to-end delay behavior (see [2, 24, 27]), to the best of our knowledge

there is no general model for per link delays. The use of a non-parametric model provides the flexibility to capture broadly different delay distributions, albeit at the cost of increasing the number of quantities to estimate (i.e. the weights in the discrete distribution). Indeed, we believe that our inference technique can shed light on the behavior and dynamics of per link delays and so provide useful results for the analysis and modeling; this we will consider in future work.

The discrete distribution can be a regarded as binned or discretized version of the (possibly continuous) true delay distribution. Use of a discrete rather than a continuous distribution allows us to perform the calculations for inference using only algebra. Formally, there is no difficulty in formulating a continuous version of the inference algorithm. However, it proceeds via inversion of Laplace transforms, a procedure that is in practice implemented numerically. In the discrete approach we can explicitly trade-off the detail of the distribution with the cost of calculation; the cost is inversely proportional to the bin widths of the discrete distribution.

The principle results of the analysis are as follows. Based on the independent delay model, we derive an algorithm to estimate the per link discrete delay distributions and utilization from the measured end-to-end delay distributions. We investigate the statistical properties of the estimator, and show it to be strongly consistent, i.e., it converges to the true distribution as the number of probes grows to infinity. We show that the estimator is asymptotically normal; this allows us to compute the rate of convergence of the estimator to its true value, and to construct confidence intervals for the estimated distribution for a given number of probes. This is important because the presence of large scale routing fluctuation (e.g. as seen in the Internet; see [26]) sets a timescale within which measurement must be completed, and hence the accuracy that can be obtained when sending probes at a given rate.

We evaluated our approach through extensive simulation in two different settings. The first set used a model simulation in which packet delays obey the independence assumption of the model. We applied the inference algorithm to the end-to-end delays generated in the simulation and compared the (true) model delay distribution. We verified the convergence to the model distribution, and also the rate of convergence, as the number of probes increased.

In the second set of experiments we conducted an `ns` simulation of packets on a multicast tree. Packet delays and losses were entirely due to queueing and packet discard mechanisms, rather than model driven. The bulk of the traffic in the simulations was background traffic due to TCP and UDP traffic sources; we compared the actual and predicted delay distributions for the probe traffic. Here we found rapid convergence, although with some persistent differences with respect to the actual distributions.

These differences appear to be caused by violation of the model due to the presence of spa-

tial dependence (i.e., dependence between delays on different links). In our simulations we find that when this type of dependence occurs, it is usually between the delays on child and parent links. However, it can extend to entire paths. As far as we know there are no experimental results concerning the magnitude of such dependence in real networks. In any case, by explicitly introducing spatial correlations into the model simulations, we were able to show that small violations of the independence assumption lead to only small inaccuracies of the estimated distribution. This continuity property of the deformation in inference due to correlations is also to be expected on theoretical grounds.

We also verified the presence of temporal dependence, i.e., dependence between the delays between successive probes on the same link. This is to be expected from the phenomenology of queueing: when a node is idle, many consecutive probes can experience constant delay; during congestion, probes can experience the same delay if their interarrival time is smaller than the congestion timescale. This poses no difficulty as all that is required for consistency of the estimator is ergodicity of the delay process, a far weaker assumption than independence. However, dependence can decrease the rate of convergence of the estimators. In our experiments, inferred values closely tracked the actual ones despite the presence of temporal dependence.

Implementation Requirements Since the data for delay inference comprises one-way packet delays, the primary requirement is the deployment of measurement hosts with synchronized clocks. Global Positioning System (GPS) systems afford one way to achieve a synchronization to within tenths of microseconds; it is currently used or planned in several of the measurement infrastructures mentioned earlier. More widely deployed is the Network Time Protocol (NTP) [20]. However, this provides accuracy only on the order of milliseconds at best, a resolution at least as coarse as the queueing delays in practice. An alternative approach that could supplement delay measurement from unsynchronized or coarsely synchronized clocks has been developed in [28, 30, 21]. These authors propose algorithms to detect clock adjustments and rate mismatches and to calibrate the delay measurements.

Another requirement is knowledge of the multicast topology. There is a multicast-based measurement tool, `mtrace` [23], already in use in the Internet. `mtrace` reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss and rate. Presently it does not support delay measurements. A potential drawback for larger topologies is that `mtrace` does not scale to large numbers of receivers as it needs to run once for each receiver to cover the entire multicast tree. In addition, `mtrace` relies on multicast routers responding to explicit measurement queries; a feature that can be administratively disabled. An

alternative approach that is closely related to the work on multicast-based loss inference [4, 5] is to infer the logical multicast topology directly from measured probe statistics; see [31] and [7]. This method does not require cooperation from the network.

Structure of the Paper. The remaining sections of the paper are organized as follows. In Section 2 we describe the delay model and in Section 3 we derive the delay estimator. In Section 4 we describe the algorithm used to compute the estimator from data. In Section 5 we present the model and network simulations used to evaluate our approach. Section 6 concludes the paper.

2 Model & Framework

2.1 Description of the Logical Multicast Tree

We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links than join them. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except for the leaf nodes and possibly the root, must have 2 or more children. We can construct the logical tree from the physical tree by deleting all links with one child (except for the root) and adjusting the links accordingly by directly joining its parent and child.

Let $\mathcal{T} = (V, L)$ denote the *logical* multicast tree, consisting of the set of nodes V, including the source and receivers, and the set of links L, which are ordered pairs (j, k) of nodes, indicating a link from j to k. We will denote $U = V \setminus \{0\}$. The set of *children* of node j is denoted by $d(j)$; these are the nodes whose parent is j. Nodes are said to be siblings if they have the same parent. For each node j, other than the root 0, there is a unique node $f(j)$, the *parent* of j, such that $(f(j), j) \in L$. Each link can therefore be also identified by its "child" endpoint. We shall define $f^n(k)$ recursively by $f^n(k) = f(f^{n-1}(k))$ with $f^1 = f$. We say that j is a descendant of k if $k = f^n(j)$ for some integer $n > 0$, and write the corresponding partial order in V as $j \prec k$. For each node j we define its *level* $\ell(j)$ to be the non-negative integer such that $f^{\ell(j)}(j) = 0$. The root $0 \in V$ represents the source of the probes and the set of *leaf* nodes $R \subset V$ (i.e., those with no children) represents the receivers.

2.2 Modeling Delay and Loss of Probe Packets

Probe packets are sent down the tree from the root node 0. Each probe that arrives at node k results in a copy being sent to every child of k. We associate with each node k a random variable D_k taking values in the extended positive real line $\mathbb{R}_+ \cup \{\infty\}$. By convention $D_0 = 0$. D_k is the random delay that would be encountered by a packet attempting to traverse the link $(f(k), k) \in L$. The value $D_k = \infty$ indicates that the packet is lost on the link. We assume that the D_k are independent. The delay experienced on the path from the root 0 to a node k is $Y_k = \sum_{j \succeq k} D_j$. Note that $Y_k = \infty$ iff $D_j = \infty$ for some $j \succeq k$, i.e. if the packet was lost on some link between node 0 and k.

Unless otherwise stated, we will discretize each link delay D_k to a set $\{0, q, 2q, \ldots, i_{\max}q, \infty\}$. Here q is the bin width, $i_{\max} + 1$ is the number of bins, and the point ∞ is interpreted as "packet lost" or "encountered delay greater than $i_{\max}q$". The distribution of D_k is denoted by α_k, where $\alpha_k(i) = \mathsf{P}[D_k = iq]$ with $\alpha_k(\infty)$ the probability that $D_k = \infty$. For each link, we denote u_k the *link utilization*; then, $u_k = 1 - \alpha_k(0)$, the probability that a packet experience delay or it is lost in traversing link k.

For each $k \in V$, the cumulative delay process Y_k, $k \in V$, takes values in $\{0, q, 2q, \ldots, i_{\max}q\ell(k), \infty\}$, i.e., it supports addition in the ranges of the constituent D_j. We set $A_k(i) = \mathsf{P}[Y_k = iq]$ with $A_k(\infty)$ the probability that $Y_k = \infty$. Because of delay independence, for finite i, $A_k(i) = \sum_{j=0}^{i} \alpha_k(j) A_{f(k)}(i-j)$, $k \in U$; by convention $A_0(0) = 1$.

We consider only **canonical delay trees**. A delay tree consists of the pair $(\mathcal{T}, \alpha)$, $\mathcal{T} = (V, L)$, $\alpha = (\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}}$. A delay tree is said to be *canonical* if $\alpha_k(0) > 0$, $\forall k \in U$, i.e., if there is a non-zero probability that a probe experiences no delay in traversing each link.

3 Delay Distribution Estimator and its Properties

Consider an experiment in which n probes are sent from the source node down the multicast tree. As result of the experiment we collect the set of source-to-leaf delays $(Y_{k,l})_{k \in R, l=1,\ldots,n}$. Our goal is to infer the internal delay characteristics solely from the collected end-to-end measurements.

In this section we state the main analytic results on which inference is based. In Section 3.1 we establish the key property underpinning our delay distribution estimator, namely the one-to-one correspondence between the link delay distributions and the probabilities of a well defined set of observable events. Applying this correspondence to measured leaf delays allows us to obtain an estimate of the link delay distribution. We show that the estimator is strongly consistent and asymptotically normal. In Section 3.2 we present the proof of the main result which also provides

the construction of the algorithm to compute the estimator we present in Section 4. In Section 3.4 we analyze the rate of convergence of the estimator as the number of probes increase.

3.1 The Delay Distribution Estimator

Let $\mathcal{T}(k) = (V(k), L(k))$ denote the subtree rooted at node k and $R(k) = R \cap V(k)$ the set of receivers which descend from k. Let $\Omega_k(i)$ denote the event $\{\min_{j \in R(k)} Y_j \leq iq\}$ that the end-to-end delay is no greater than iq for at least least one receiver in $R(k)$. Let $\gamma_k(i) = \mathsf{P}[\Omega_k(i)]$ denote its probability. Finally let Γ denote the mapping associating the link distributions $(\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}}$ to the probabilities of the events $\Omega_k(i)$, $\gamma = (\gamma_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}}$. The proof of the next result is given in the following section.

Theorem 1 *Let* $\mathcal{A} = \{\alpha = (\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}} : \alpha_k(0) > 0, \ \sum_{i \leq i_{\max}} \alpha_k(i) \leq 1\}$ *and* $\mathcal{G} = \{\gamma = (\gamma_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}} : \exists \alpha \in \mathcal{A} | \gamma = \Gamma(\alpha)\}$. Γ *is a bijection from* $\mathcal{A}$ *to* $\mathcal{G}$ *which is continuously differentiable and has a continuously differentiable inverse.*

Estimate γ by the empirical probabilities $\widehat{\gamma}$, where

$$\widehat{\gamma}_k(i) = n^{-1} \sum_{m=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k,m} \leq iq\}}, \tag{1}$$

$\mathbf{1}_{\{S\}}$ denotes the indicator function of the set S and $(\widehat{Y}_{k,m})_{k \in U, m=1,\ldots,n}$ are the subsidiary quantities

$$\widehat{Y}_{k,m} = \min_{d \in R(k)} \widehat{Y}_{d,m}, \quad k \in U. \tag{2}$$

Our estimate of $\alpha_k(i)$ is $\widehat{\alpha}_k(i) = (\Gamma^{-1}(\widehat{\gamma}))_k(i)$. We estimate link k utilization by $\widehat{u}_k = 1 - \widehat{\alpha}_k(0)$.

Let $\mathcal{A}^{(1)} = \{\alpha = (\alpha_k(i))_{k \in U, i \in \{0,\ldots,i_{\max}\}} : \alpha_k(0) > 0, \ \sum_{i \leq i_{\max}} \alpha_k(i) < 1\}$ denote the open interior of $\mathcal{A}$. The following holds:

Theorem 2 *When* $\gamma \in \Gamma(\mathcal{A}^{(1)})$, *as* $n \to \infty$, $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ *converge almost surely to* α, *i.e., the estimator is strongly consistent.*

Proof: Since Γ^{-1} is continuous on $\Gamma(\mathcal{A}^{(1)})$ and $\mathcal{A}^{(1)}$ is open in $\mathcal{A}$, it follows that $\Gamma(\mathcal{A}^{(1)})$ is an open set in $\Gamma(\mathcal{A})$. By the Strong Law of large numbers, since $\widehat{\gamma}$ is the mean of n independent random variables, $\widehat{\gamma}$ converges to γ almost surely for $n \to \infty$. Therefore, when $\gamma \in \Gamma(\mathcal{A}^{(1)})$, there exists n_0 such that $\widehat{\gamma} \in \Gamma(A^{(1)})$, $n > n_0$. Then, the continuity of Γ^{-1} insures that $\widehat{\alpha}$ converges almost surely to α as $n \to \infty$. ∎

3.2 Proof of Theorem 1

To prove the Theorem, we first express γ as function of α and then show that the mapping from $\mathcal{A}$ to $\mathcal{G}$ is injective.

3.2.1 Relating γ to α

Denote $\beta_k(i) = \mathsf{P}[\min_{j \in R(k)} Y_j - Y_{f(k)} \leq iq]$, $i = 0, \cdots, i_{\max}$. $\beta_k(i)$ obeys the recursion

$$\begin{array}{ll} \beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) \left[1 - \prod_{d \in d(k)} (1 - \beta_d(i-j))\right] & k \in U \setminus R \\ \beta_k(i) = \sum_{j=0}^{i} \alpha_k(j) & k \in R. \end{array} \tag{3}$$

Then, by observing that

$$\gamma_k(i) = \sum_{j=0}^{i} \beta_k(i-j) A_{f(k)}(j), \tag{4}$$

$k \in U \setminus R$, we readily obtain

$$\begin{array}{ll} \gamma_k(i) = \sum_{j=0}^{i} A_k(j) \left[1 - \prod_{d \in d(k)} (1 - \beta_d(i-j))\right] & k \in U \setminus R \\ \gamma_k(i) = \sum_{j=0}^{i} A_k(j) & k \in R \end{array} \tag{5}$$

The set of equations (5) completely identifies the mapping Γ from $\mathcal{A}$ to $\mathcal{G}$. The mapping is clearly continuously differentiable. Observe that the above expressions can be regarded as a generalization of those derived for the loss estimator in [4] (by identifying the event *no delay* with the event *no loss*).

3.2.2 Relating α to γ

It remains to show that the mapping from $\mathcal{A}$ to $\mathcal{G}$ is injective. To this end, below we derive an algorithm for inverting (5). We postpone to Appendix A the proof that the inverse is unique and continuously differentiable. For sake of clarity we separate the algorithm into two parts: in the first we derive the cumulative delay distributions A from γ; then, we deconvolve A to obtain α.

Computing A

Step 0:

Solve (5) for $i = 0$. This amounts solving the equation

$$(1 - \gamma_k(0)/A_k(0)) = \prod_{d \in d(k)} (1 - \gamma_d(0)/A_k(0)),\ k \in U \setminus R \tag{6}$$

and

$$\gamma_k(0) = A_k(0),\ k \in R. \tag{7}$$

This equation is formally identical to the one of the loss estimator [4]. From [4], we have that the solution of (6) exists and is unique in $(0,1)$ provided that $0 < \gamma_k(0) < \sum_{d \in d(k)} \gamma_d(0)$ which holds for canonical delay trees. We then compute $\beta_k(0) = \gamma_k(0)/A_{f(k)}(0)$, $k \in U$.

Step i:

Given $A_k(j)$ and $\beta_k(j)$, $k \in U$, $j = 0, \ldots, i-1$, in this step we compute $A_k(i)$ and $\beta_k(i)$, $k \in U$. For $k \in U \setminus R$, in expression (5) we replace $\beta_d(i)$ with $\frac{\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j) A_k(j) - \beta_d(0) A_k(i)}{A_k(0)}$ (from (4)) and obtain the following equation

$$\begin{array}{c} \gamma_k(i) + A_k(0) \left\{ \prod_{d \in d(k)} \left[1 - \frac{\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j) A_k(j) - \beta_d(0) \boldsymbol{A_k(i)}}{A_k(0)} \right] - 1 \right\} + \\ \sum_{j=1}^{i-1} A_k(j) \left\{ \prod_{d \in d(k)} \left[1 - \beta_d(i-j) \right] - 1 \right\} + \boldsymbol{A_k(i)} \left\{ \prod_{d \in d(k)} \left[1 - \beta_d(0) \right] - 1 \right\} = 0 \end{array} \tag{8}$$

(the unknown term $A_k(i)$ is highlighted in boldface). This is a polynomial in $A_k(i)$ of degree $\#d(k)$. As shown in Appendix A we consider the second largest solution of (8).

For $k \in R$, we directly compute $A_k(i)$ from (5), $A_k(i) = \gamma_k(i) - \sum_{j=0}^{i-1} A_k(j)$. Then we compute $\beta_k(i)$, $k \in U$, as $\beta_k(i) = \frac{\gamma_k(i) - \sum_{j=1}^{i} A_{f(k)}(j) \beta_k(i-j)}{A_{f(k)}(0)}$.

Computing α

Once step $i_{\max}$ is completed, we compute $\alpha_k(i)$, $k \in U$ as follows

$$\alpha_k(i) = \begin{cases} \frac{A_k(0)}{A_{f(k)}(0)} & i = 0 \\ \frac{A_k(i) - \sum_{j=1}^{i} A_{f(k)}(j) \alpha_k(i-j)}{A_{f(k)}(0)} & i = 1, \cdots, i_{\max}. \end{cases} \tag{9}$$

3.3 Example: the Two-leaf Tree

In this section we illustrate the application of the results of Section 3.1 to the two-leaf tree of Figure 1. We assume that on each link, a probe either suffers no delay, a unit amount of delay, or is otherwise lost; for $k \in \{1,2,3\}$, therefore, delay takes values in $\{0, 1, \infty\}$.

For this example, equations (6) and (8) can be solved explicitly; combined with (9) we obtain

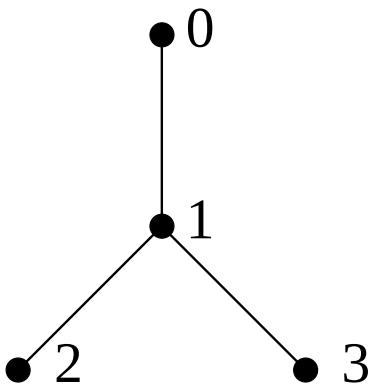


Figure 1: TWO-LEAF MULTICAST TREE.

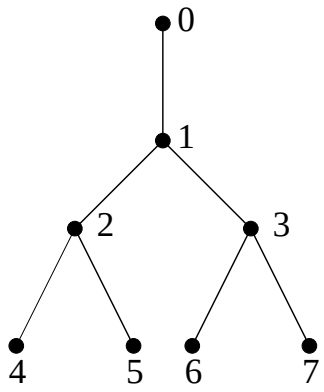


Figure 2: FOUR-LEAF MULTICAST TREE.

the estimates

$$
\begin{aligned}
\widehat{\alpha}_1(0) &= \tfrac{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}{\widehat{\gamma}} \\
\widehat{\alpha}_2(0) &= \tfrac{\widehat{\gamma}}{\widehat{\gamma}_3(0)} \\
\widehat{\alpha}_3(0) &= \tfrac{\widehat{\gamma}}{\widehat{\gamma}_2(0)} \\
\widehat{\alpha}_1(1) &= \tfrac{1}{2}\tfrac{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}{\widehat{\gamma}}\left(\tfrac{\widehat{\gamma}_2(1)}{\widehat{\gamma}_2(0)}+\tfrac{\widehat{\gamma}_3(1)}{\widehat{\gamma}_3(0)}-1-\sqrt{\left(\tfrac{\widehat{\gamma}_2(1)}{\widehat{\gamma}_2(0)}+\tfrac{\widehat{\gamma}_3(1)}{\widehat{\gamma}_3(0)}-1\right)^2-4\tfrac{\widehat{\gamma}_2(1)\widehat{\gamma}_3(1)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}+4\tfrac{\widehat{\gamma}_2(1)+\widehat{\gamma}_3(1)-\widehat{\gamma}_1(1)}{\widehat{\gamma}}}\right) \\
\widehat{\alpha}_2(1) &= \tfrac{\widehat{\gamma}_2(1)-\widehat{\gamma}_2(0)-\widehat{\alpha}_1(1)\widehat{\gamma}/\widehat{\gamma}_3(0)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}\widehat{\gamma} \\
\widehat{\alpha}_3(1) &= \tfrac{\widehat{\gamma}_3(1)-\widehat{\gamma}_2(0)-\widehat{\alpha}_1(1)\widehat{\gamma}/\widehat{\gamma}_2(0)}{\widehat{\gamma}_2(0)\widehat{\gamma}_3(0)}\widehat{\gamma}
\end{aligned}
$$

where $\widehat{\gamma} = \widehat{\gamma}_2(0) + \widehat{\gamma}_3(0) - \widehat{\gamma}_1(0)$.

3.4 Rates of Convergences of the Delay Distribution Estimator

3.4.1 Asymptotic Behavior of the Delay Distribution Estimator

In this section, we study the rate of convergence of the estimator. Theorem 2 states that $\widehat{\alpha}$ converges to α with probability 1 as n grows to infinity; but it provides no information on the rate of convergence. Because of the mild conditions satisfied by Γ^{-1}, we can use Central Limit Theorem to establish the following asymptotic result

Theorem 3 *When $\gamma \in \Gamma(\mathcal{A}^{(1)})$, as $n \to \infty$, $\sqrt{n}(\widehat{\alpha}-\alpha)$ converges in distribution to a multivariate normal random variable with mean vector 0 and covariance matrix $\nu = D(\alpha)\cdot\sigma\cdot D^T(\alpha)$ where $\sigma_{(k_1,i)(k_2,j)} = \lim_{n\to\infty} n\mathsf{Cov}(\widehat{\gamma}_{k_1}(i),\widehat{\gamma}_{k_2}(j))$, for $k_1, k_2 \in U$, $i,j \in \{0,\ldots,i_{\max}\}$, $D_{(k_1,i)(k_2,j)}(\alpha) = \frac{\partial\Gamma^{-1}_{k_1}(i)}{\partial\gamma_{k_2}(j)}(\Gamma(\alpha))$ and D^T denotes the transpose.*

Proof: By the Central Limit Theorem, it follows that the random variables $\widehat{\gamma}$ are asymptotically Gaussian as $n \to \infty$ with

$$\sqrt{n}(\widehat{\gamma} - \gamma) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma)$$

Here $\mathcal{D}$ denotes convergence in distribution. Following the same lines of the proof of Theorem 1, when $\gamma \in \Gamma(\mathcal{A}^{(1)})$, there existst n_0 such that $\widehat{\gamma} \in \Gamma(\mathcal{A}^{(1)})$, $n > n_0$. Then, Since Γ^{-1} is continuously differentiable on $\mathcal{G}$, the Delta method (see Chapter 7 of [34]) yields that $\widehat{\alpha} = \Gamma^{-1}(\widehat{\gamma})$ is also asymptotically Gaussian as $n \to \infty$:

$$\sqrt{n}(\Gamma^{-1}(\widehat{\gamma}) - \alpha) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \nu)$$

■

Theorem 3 allows us to compute confidence intervals of the estimates, and therefore their accuracy and their convergence rate to the true values as n grows. This is relevant in assessing: (i) the number of probes required to obtain a desired level of accuracy of the estimate; (ii) the likely accuracy of the estimator from actual measurements by associating confidence intervals to the estimates.

For large n, the estimator $\widehat{\alpha}_k(i)$ will lie in the interval

$$\alpha_k(i) \pm z_{\delta/2} \sqrt{\frac{\nu_{(k,i)(k,i)}}{n}}, \tag{10}$$

where $z_{\delta/2}$ is the $1 - \delta/2$ quantile of the standard distribution and the interval estimate is a $100(1 - \delta)\%$ confidence interval.

To obtain the confidence interval for $\widehat{\alpha}$ derived from measured data from n probes, we estimate ν by $\widehat{\nu} = D(\widehat{\alpha}) \cdot \widehat{\sigma} \cdot D^T(\widehat{\alpha})$ where

$$\widehat{\sigma}_{(k_1,i)(k_2,j)} = \frac{1}{n-1} \left(\sum_{l=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k_1,l} \le id \ \wedge \ \widehat{Y}_{k_2,l} \le jd\}} - \frac{1}{n} \sum_{l=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k_1,l} \le id\}} \sum_{l=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k_2,l} \le jd\}} \right),$$

and $D(\widehat{\alpha})$ is the Jacobian of the inverse map Γ^{-1} computed for $\alpha = \widehat{\alpha}$. We then use confidence intervals of the form

$$\alpha_k(i) \pm z_{\delta/2} \sqrt{\frac{\widehat{\nu}_{(k,i)(k,i)}}{n}}. \tag{11}$$

3.4.2 Dependence of the Delay Distribution Estimator on Topology

The estimator variance determines the number of probes required to obtain a given level of accuracy. Therefore, it is important to understand how the variance is affected by the underlying

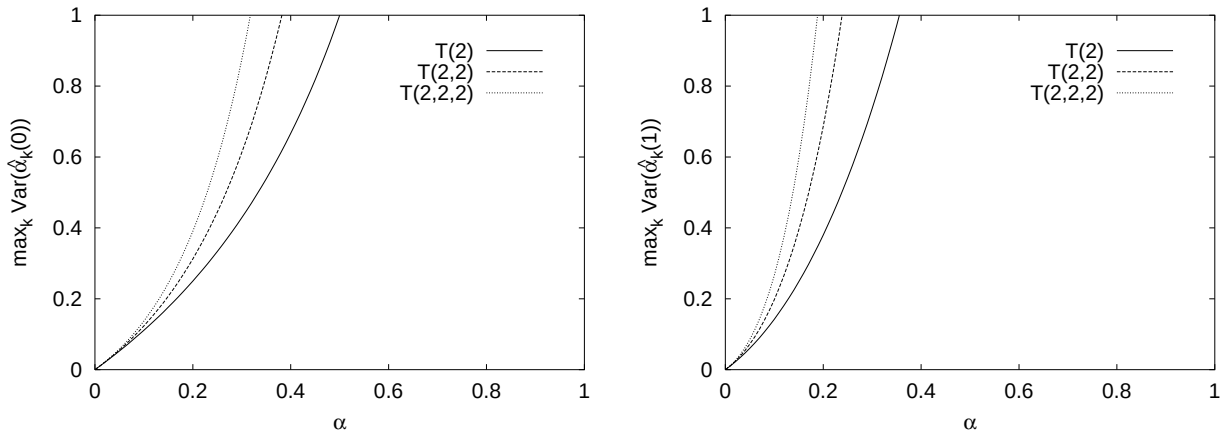


Figure 3: ASYMPTOTIC ESTIMATOR VARIANCE AND TREE DEPTH. Binary tree with depth 2, 3 and 4. Left: Minimum and Maximum Variance of the estimates $\widehat{\alpha}_k(0)$ (a) and $\widehat{\alpha}_k(1)$ (b) over all links.

parameters, namely the delay distributions and the multicast tree topology. The following Theorem, the proof of which we postpone to Appendix C, characterizes the behavior of the variance for small delays. Set $\| \alpha \| = \max_{k \in U, i>0} \alpha_k(i)$.

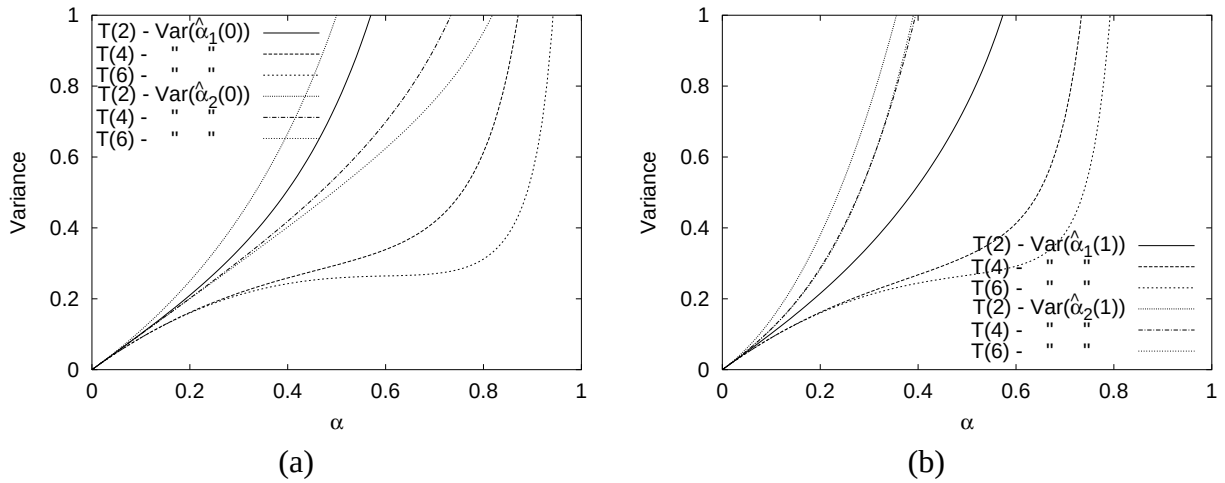
Theorem 4 *As* $\| \alpha \| \to 0$,

$$\nu = \begin{pmatrix} \nu_{k_1 k_1} & 0 & 0 & \dots & 0 \\ 0 & \nu_{k_2 k_2} & 0 & \dots & 0 \\ 0 & 0 & \nu_{k_3 k_3} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & \nu_{k_{\#U} k_{\#U}} \end{pmatrix} + O(\| \alpha \|^2) \text{ with } \nu_{kk} = \begin{pmatrix} \sum_{i>0} \alpha_k(i) & \alpha_k(1) & \alpha_k(2) & \dots & \alpha_k(i_{\max}) \\ \alpha_k(1) & \alpha_k(1) & 0 & \dots & 0 \\ \alpha_k(2) & 0 & \alpha_k(2) & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha_k(i_{\max}) & 0 & 0 & \dots & \alpha_k(i_{\max}) \end{pmatrix} \tag{12}$$

.

Theorem 4 states that the estimator variance is, to first order, independent of the topology. To explore higher order dependencies, we computed the asymptotic variance for a selection of trees with different depths and branching ratio. We use the notation $T(r_1, \dots, r_m)$ to denote a tree of $m+1$ levels where, apart from node 0 that has one descendent, nodes at level j have exactly r_j children. For simplicity, we consider the case when link delay takes values in $\{0, 1\}$, *i.e.*, we consider no loss, and study the behavior as function of $\alpha_k(1) = \alpha$.

In Figure 3 we show the dependence on tree depth for binary trees of depth 2, 3 and 4. We plot the maximum value of the variance over the links $\max_k \mathsf{Var}(\widehat{\alpha}_k(0))$ (a) and $\max_k \mathsf{Var}(\widehat{\alpha}_k(1))$ (b). In these examples, the variance increases with the tree depth. In Figure 4 we show the dependence



(a) (b)

Figure 4: ASYMPTOTIC ESTIMATOR VARIANCE AND BRANCHING RATIO. Binary tree with depth 2 and 2, 4 or 6 receivers. Left: Variance of $\widehat{\alpha}_k(0)$ (a) and $\widehat{\alpha}_k(1)$ (b) for link 1 (common link) and 2 (generic receiver).

on branching ratio for a tree of level 2. We plot the estimator variance for both link 1 (the common link) and link 2 (a generic receiver). In these examples, increasing the branching ratio decreases the variances, especially those of the common link estimates which increases less than linearly for α up to 0.7 when the branching ratio is larger than 3. In all cases, the variance of $\widehat{\alpha}_k(1)$ is larger than $\widehat{\alpha}_k(0)$.

In all cases, as predicted by Theorem 4, the estimator variance is asymptotically linear in α independently of the topology as $\alpha \to 0$. As α increases, the behavior is affected by different factors: increasing the branch ratio results in a reduction of the variance, while increasing the tree depth results in variance increase. The first can be explained in terms of the increased number of measurements available for the estimation as the number of receivers sharing a given link increases; the second appears to be the effect of cumulative errors that accrue as the number of links along a path increases (α is computed iteratively on the tree). We also observe that the variance increases with the delay lag; this appears to be caused by the iterative computation on the number of bins that progressively cumulate errors.

4 Computation of the Delay Distribution Estimator

In this section we describe an algorithm for computing the delay distribution estimate from measurements based on the results presented in the previous section. We also discuss its suitability for distributed implementation and how to adapt the computation to handle different bin sizes.

We assume the experimental data of source-to-leaf delays $(Y_{k,m})_{k\in R,m=1,\ldots n}$ from n probes, as collected at the leaf nodes $k \in R$. Two steps must be initially performed to render the data into a form suitable for the inference algorithms: (i) removal of fixed delays and (ii) choosing a bin size q and computing the estimate $\widehat{\gamma}$.

The first step is necessary since it is generally not possible to apportion the deterministic component of the source-to-leaf delays between interior links. (To see this, it is sufficient to consider the case of the two receiver tree; expressing the link fixed delays in terms of the source-to-leaf fixed delays results in two equations in three unknowns). Thus we normalize each measurement by subtracting the minimum delay seen at the leaf. Observe that to interpret the observed minimum delay as the transmission delay assumes that at least one probe has experienced no queuing delay along the path).

The second step is to choose the bin size q and discretize the delays measurements accordingly. This introduces a quantization error which affects the accuracy of the estimates. As our results have shown, the accuracy increases as q decreases (we have obtained accurate results over a significant range of values of q up to the same order of magnitude of the links average delay). The choice of q represents a trade-off between accuracy and cost of the computation as a smaller bin size entails a higher computational cost due to the higher dimensionality of the binned distributions.

These two steps are carried out as follows. From the measured data $(Y_k)_{k\in R}$, we recursively construct the auxiliary vector process $\widehat{Y} = (\widehat{Y}_k)_{k\in V}$

$$\widehat{Y}_{k,l} = Y_{k,l} - \min_{m\in\{1,\ldots,n\}} Y_{k,m},\ k \in R \tag{13}$$

$$\widehat{Y}_{k,l} = \min_{j\in d(k)} \widehat{Y}_{j,l},\ k \in V \setminus R. \tag{14}$$

The binned estimates of $\widehat{\gamma}$ are

$$\widehat{\gamma}_k(i) = n^{-1} \sum_{m=1}^{n} \mathbf{1}_{\{\widehat{Y}_{k,m} \le iq+q/2]\}},\ i = 0, \ldots, i_{\max}, \tag{15}$$

with

$$i_{\max} = \left\lceil \frac{\max_{k\in R} \max_{m\in N_k(n)} \widehat{Y}_{k,m}}{q} \right\rceil.$$

Here $\lceil x \rceil$ denotes the smallest integer greater than x and $N_k(n) = \{m \in \{1, \ldots, n\} | Y_{k,m} < \infty\}$. Observe that i_{max} represents the largest value at which the estimate $\widehat{\alpha}(i_{\max})$ is non zero.

The estimate can be computed iteratively over the delay lag and recursively over the tree. The pseudo code for carrying out the computation is found in Figure 5. The procedure `find_y` calculates $\widehat{Y}_k$ and $\widehat{\gamma}_k$, with $\widehat{Y}_{k,l}$ initialized to $Y_{k,l} - \min_{m\in\{1,\ldots,n\}} Y_{k,m}$ for $k \in R$ and ∞ (a value

```
procedure main {
     find_y ( 1 ) ;
     foreach ( i ∈ {0, ···, i_max} )
          infer_delay ( 1 , i ) ;
}

procedure find_y ( k ) {
     foreach ( j ∈ d(k) ) {
          Ŷ_j = find_y ( j ) ;
          foreach ( m ∈ {1, ..., n} )
               Ŷ_k[m] = min{Ŷ_k[m], Ŷ_j[m]};
     }
     foreach ( i ∈ {0, ···, i_max} )
          γ̂_k[i] = n^{-1} Σ_{j=1}^{n} 1_{{Ŷ_k[j] ≤ iq+q/2}};
     return Ŷ_k ;
}

procedure infer_delay ( k, i ) ;
     if ( i == 0 ) {
          Â_k[i] = solvefor1 ( Â_k[i] , (1 − γ̂_k[i]/Â_k[i] == ∏_{d∈d(k)} 1 − γ̂[i]/Â[i] ) ;
     } else {
          Â_k[i] = solvefor2 ( Â_k[i] , γ̂_k[i] + Â_k[0] { ∏_{d∈d(k)} [ 1 − (γ̂_d[i] − Σ_{j=1}^{i} β̂_d[i−j] Â_k[j]) / Â_k[0]) ] − 1 } +
          Σ_{j=1}^{i} Â_k[j] { ∏_{d∈d(k)} [ 1 − β̂_d[i − j] ] − 1 } == 0 ) ;
     }
     β̂_k[i] = (γ̂_k[i] − Σ_{j=1}^{i} Â_{f(k)}[j] β̂_k[i−j]) / Â_{f(k)}[0] ;
     α̂_k[i] = (Â_k[i] − Σ_{j=1}^{i} Â_{f(k)}[j] α̂_k[i−j]) / Â_{f(k)}[0] ;
     foreach ( j ∈ d(k) )
          infer_delay ( j , i ) ;
}
```

Figure 5: PSEUDOCODE FOR INFERENCE OF DELAY DISTRIBUTION.

larger than any observed delay suffices) otherwise. The procedure `infer_delay` calculates $\widehat{\alpha}_k(i)$ for a fixed i recursively on the tree, with $\widehat{A}_k[i]$, $k \in V$, $i = 0, \cdots, i_{\max}$ initialized to 0, except for $\widehat{A}_0[0]$ set to 1. The output of the algorithm are the estimates $\widehat{\alpha}_k$, $k \in U$.

Within the code, an empty product (which occurs when the first argument of `infer` is a leaf) is assumed to be zero. The routines `solvefor1` and `solvefor2` return the value of the first symbolic argument that solves the equation in the second argument. `solvefor1` returns a solution in $(0, 1)$; from Lemma 1 in [4] this is known to be unique. `solvefor2` returns the unique solution if the second argument is linear in $\widehat{A}_k(i)$ (this happen only if k is a leaf-node), otherwise it returns the second largest solution.

4.1 Distributed Implementation

As with the loss estimator [4] the algorithm is recursive on trees. In particular, observe that the computation of $\widehat{\gamma}$ and $\widehat{A}_k$ only requires the knowledge of $(\widehat{Y}_{j,m})_{j\in d(k),m=1,\ldots,n}$; these are computed recursively on the the tree starting from the receivers. Therefore it is possible to distribute the computation among the nodes of the tree (or representative nodes of subtrees), with each node k being responsible for the aggregation of the measurements of its child nodes through (14) and for the computation of $\widehat{A}_k$.

4.2 Adopting Different Bin Sizes

Following the results of the previous section, we presented the algorithm using a fixed value of q for all links. This can be quite restrictive in a heterogeneous environment, where links may differ significantly in terms of speed and buffer sizes; a single value of q could be at the same time too coarse grained for describing the delay of a high bandwidth link but too fine-grained to efficiently capture the essential characteristics of the delay experienced along a low bandwidth link.

A simple way to overcome this limitation is to run the algorithm for different values of q, each best suited for the behavior of a different group of links, and retain each time only the solutions for those links. A drawback of this approach is that each distribution is computed for all the different bin sizes. The distributed nature of the algorithm suggests we can do better; indeed, since A_k, $k \in U$, can be computed independently from one another, it is possible to compute each link delay distribution only for the bin size best suited to its delay characteristics. More precisely, let q_k denote the bin size adopted for link k. In order to compute $\widehat{\alpha}_k$ with bin size q_k we need to compute both $\widehat{A}_k$ and $\widehat{A}_{f(k)}$ with bin size q_k. Thus, the overall computation requires calculating each cumulative distribution $\widehat{A}_k$ only for the bin sizes q_j, $j \in d(k) \cup \{k\}$, *i.e.*, only for the bin sizes adopted for the links terminating at node k and at all its child nodes rather than for bin sizes adopted for all links.

In an implementation, we envision that a fixed value for all links is used first. This can be chosen based on the measurements spread and the tree topology or delay past history. Then, with a better idea of each link delay spread, it would be possible to refine the value of the bin size on a link by link basis.

5 Experimental Evaluation

We evaluated our delay estimator through extensive simulation. Our first set of experiments focus on the statistical properties of the estimator. We perform *model simulation*, where delay and loss are determined by random processes that follow the model on which we based our analysis. In

our second set of experiments we we investigate the behavior of the estimators in a more realistic setting where the model assumption of independence may be violated. To this end, we perform *TCP/UDP simulation*, using the `ns` simulator. Here delay and loss are determined by queuing delay and queue overflows at network nodes as multicast probes compete with traffic generated by TCP/UDP traffic sources.

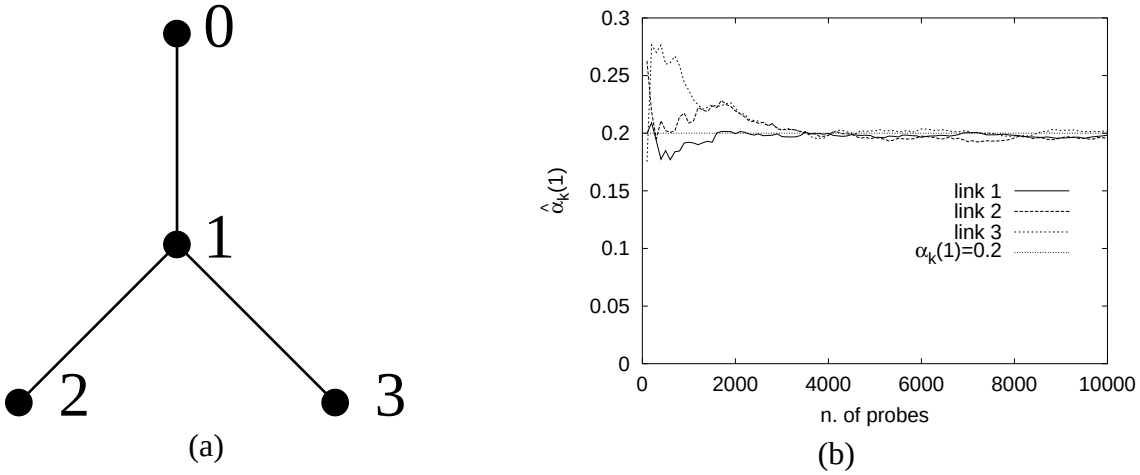
5.1 Comparing Inferred vs. Sample Distributions

Before examining the results of our experiments, we describe our approach to assessing the accuracy of the inferred distributions. Given an experiment in which n probes are sent from the source to the receivers, for $k \in V$, the inferred distribution $\widehat{\alpha}_k$ ($\widehat{A}_k$) is computed from the end-to-end measurements using the algorithm described in Section 4. Its accuracy must be measured against the actual data, represented by a finite sequence of delays $\{D_{k,m}\}_{m=1}^{n}$ ($\{Y_{k,m}\}_{m=1}^{n}$), experienced by the probes in traversing (reaching) that link. For simplicity of notation we assume, hereafter, that each set of data has been already normalized by subtracting the minimum delay from the sequence.

We compare summary statistics of link delay, namely the mean and the variance. A finer evaluation of the accuracy lies in a direct comparison of the inferred and sample distributions. To this end, we also compute the largest absolute deviation between the inferred and sample c.d.f.s. This measure is used in statistics for the Kolmogoroff-Smirnoff test for goodness of fit of a theoretical with a sample distribution. A small value for this measure indicates that the theoretical distribution provides a good fit to the sample distribution; a large value leads to the rejection of the hypothesis. We cannot directly apply the test as we deal with an inferred rather than a sample c.d.f.; however, we will use the largest absolute deviation as a global measure of accuracy of the inferred distributions.

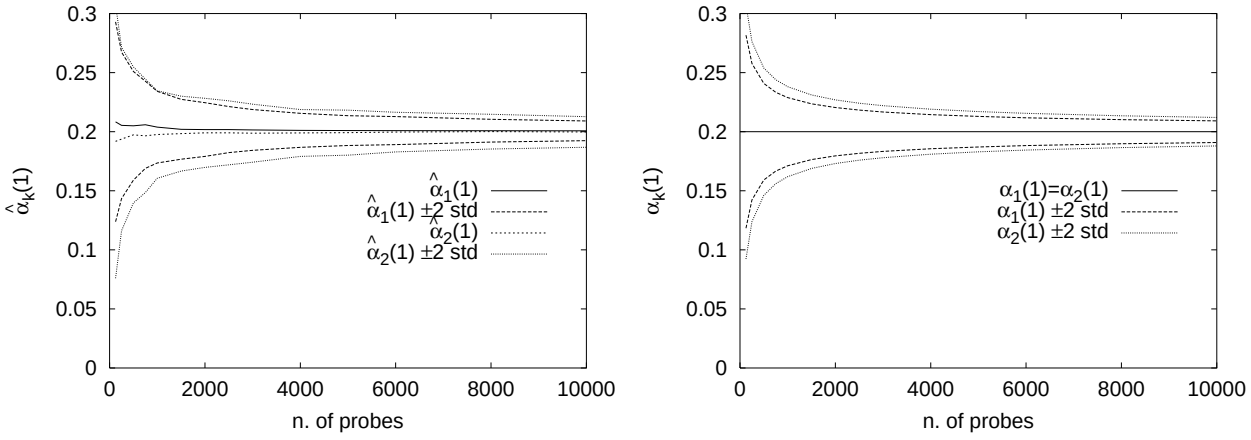
We compute the sample distributions $\tilde{\alpha}$ and $\tilde{A}$ using the same bin size q of the estimator. More precisely, we compute $(\tilde{\alpha}_k)_{k\in V}$ and $(\tilde{A}_k)_{k\in V}$ as $\tilde{\alpha}_k(i) = \#N_{f(k)}(n)^{-1} \sum_{j\in N_{f(k)}(n)} \mathbf{1}_{\{D_{k,j}\in(id-d/2,id+d/2]\}}$, $i = 0, \ldots, i_{\max}$ ($\tilde{\alpha}_k(\infty) = \#N_{f(k)}(n)^{-1} \sum_{j\in\#N_{f(k)}(n)} \mathbf{1}_{\{D_{k,j}=\infty\}}$) and $\tilde{A}_k(i) = n^{-1} \sum_{j=1}^{n} \mathbf{1}_{\{Y_{k,j}\in(id-d/2,id+d/2]\}}$, $i = 0, \ldots, i_{\max}$ ($\tilde{A}_k(\infty) = n^{-1} \sum_{j=1}^{n} \mathbf{1}_{\{Y_{k,j}=\infty\}}$). (Observe that in computing $(\tilde{\alpha}_k)_{k\in V}$, the sum is carried out only over $N_{f(k)}(n) = \{j \in \{1, \ldots, n\} | Y_{f(k)}, j < \infty\}$, the set over which the delay along link k is defined either finite or infinite.)

The largest absolute deviation between the inferred and sample c.d.f.s is, then, $\Delta_k = \max_{l=0,\ldots,i_{\max}}$ $|\sum_{i=0}^{l} \tilde{\alpha}_k(i) - \sum_{i=0}^{l} \widehat{\alpha}_k(i)|$. In other words, Δ_k is the smallest nonnegative number such that $\sum_{j\le i} \tilde{\alpha}_k(i)$ lies between $\sum_{j\le i} \widehat{\alpha}_k(i) \pm \Delta_k$ $i = 0, \ldots, i_{\max}$. The same result holds for the tail probabilities, $\sum_{j>i} \alpha_k(i)$.



(a) (b)

Figure 6: TWO RECEIVERS TREE. (a): Simulation topology. (b): Convergence of $\widehat{\alpha}_k(1)$ to $\alpha_k(1)$. $\alpha_k(0) = 0.79$, $\alpha_k(1) = 0.2$ and $\alpha_k(\infty) = 0.01$ for all links.



(a) (b)

Figure 7: AGREEMENT BETWEEN SIMULATED AND THEORETICAL CONFIDENCE INTERVALS. (a): Results from 100 model simulations. (b): Prediction from (10). The graphs show two-sided confidence interval at 2 standard deviation for link 1 and 2. Parameters are $\alpha_k(1) = 0.2$ and $\alpha_k(\infty) = 0.01$ for all links.

5.2 Model Simulation

We first consider the two-leaf topology of Figure 6(a), with source 0 and receivers 2 and 3. Link delays are independent, taking values in $\{0, 1, \infty\}$; if a probe is not lost it experiences either no delay or unit delay. In Figure 6(b) we plot the estimate $\widehat{\alpha}_k(1)$ versus the model values for a run comprising 10000 probes. The estimate converges within to 2% of the model value within 4000 probes. In Figure 7 we compare the empirical and theoretical 95% confidence intervals. The theoretical intervals are computed from (10). The empirical intervals are computed over 100 independent simulations. The agreement between simulation and theory is close: the two sets of curves are almost indistinguishable.

Next we consider the topology of Figure 8. Delays are independently distributed according to a truncated geometric distribution taking values in $\{0, 1, \ldots, 40, \infty\}$ (in ms) . This topology is also used in subsequent TCP/UDP simulations, and the link average delay and loss probability are chosen to match the values obtained from these. The average delay range between 1 and 2ms for the slower *edge* links and between 0.2 and 0.5ms for the *interior* faster links; the link losses range from 1% to 11%. In Figure 9 we plot the estimated average link delay and standard deviation with the empirical 95% confidence interval computed over 100 simulations. The results are very accurate even for several hundred probes: the theoretical average delay always lies within the confidence interval and the standard deviation does so for 1500 or more probes.

To compare the inferred and sample distributions, we computed the largest absolute deviation between the inferred and sample c.d.f.s. The results are summarized in Figure 10 where we plot the minimum, median and the maximum largest absolute deviation in 100 simulations computed over all links as a function of n (a) and link by link for $n = 10000$ (b). The accuracy increases with the number of probes as $1/\sqrt{n}$ with a spread of two orders of magnitude between the minimum and maximum. For more than 3000 probes, the average largest deviation over all links is less than 1%. The accuracy varies from link to link: when the number of probes is $n = 10000$, then at one extreme we have link 4 with $0.18\% \leq \Delta_4 \leq 0.8\%$ and at the other extreme link 6 with $0.3\% \leq \Delta_6 \leq 4\%$ over 100 simulations. We observe that the inferred distributions are less accurate as we go down the tree. This is in agreement with the results of Section 3.4 and is explained in terms of the larger inferred probabilities variances of downstream with respect to upstream nodes.

5.3 TCP/UDP Simulations

We used the topology shown in Figure 8. To capture the heterogeneity between edges and core of a WAN, interior links have higher capacity (5Mb/sec) and propagation delay (50ms) then at the

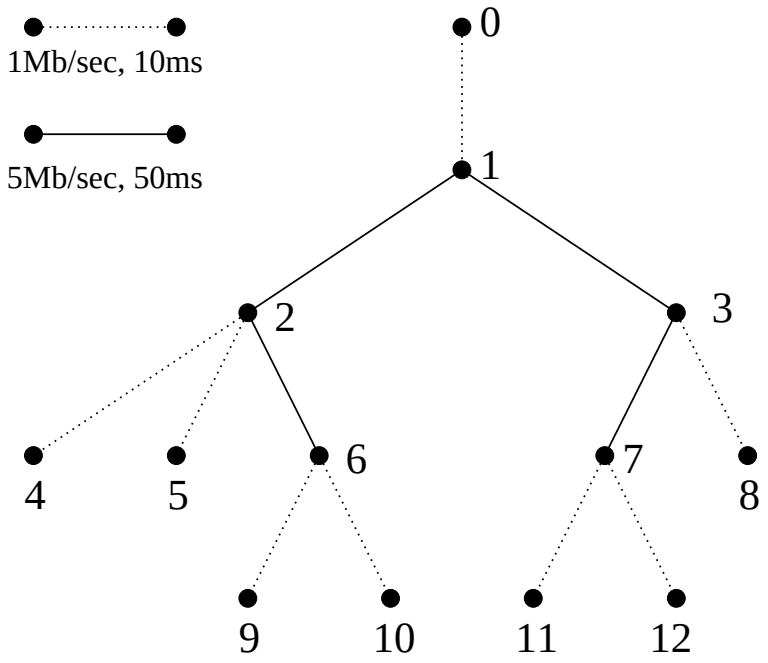
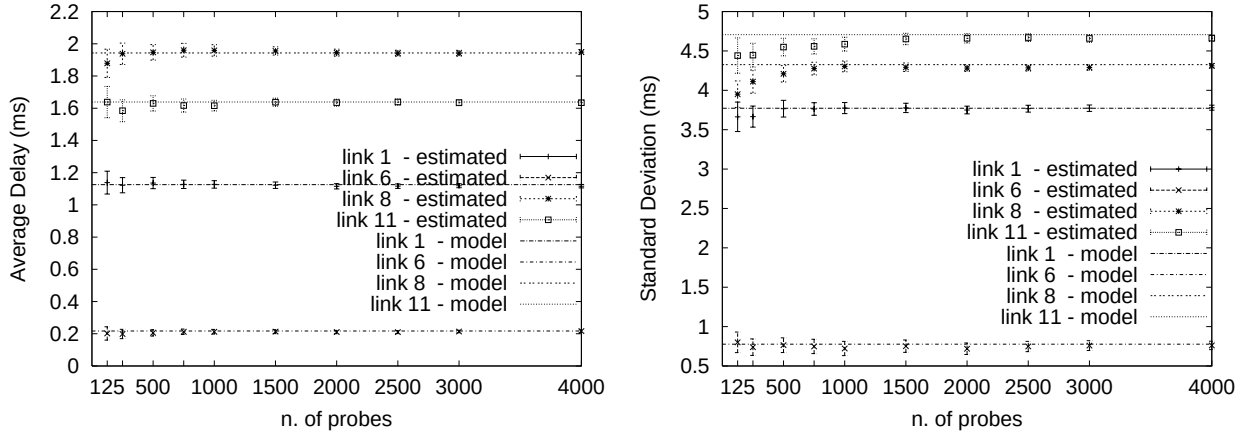


Figure 8: Simulation Topology: Link are of two types: *edge* links of 1MB/s capacity and 10ms latency, and *interior* links of 5Mb/s capacity and 50ms latency.



(a) (b)

Figure 9: MODEL SIMULATION: TOPOLOGY OF FIGURE 8. ESTIMATED VERSUS THEORETICAL DELAY AVERAGE AND STANDARD DEVIATION WITH 95% CONFIDENCE INTERVAL COMPUTED OVER 100 MODEL SIMULATIONS.

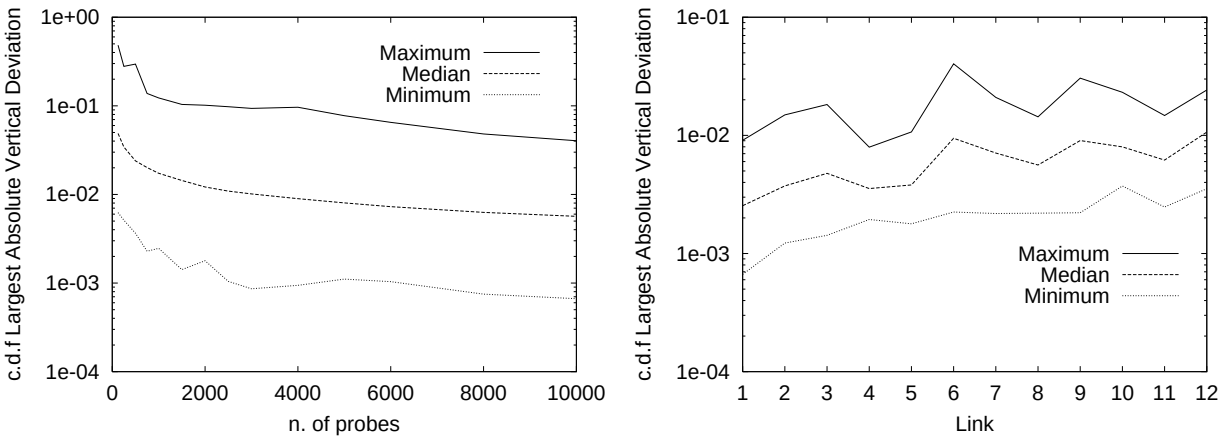


Figure 10: MODEL SIMULATION: TOPOLOGY OF FIGURE 8. ACCURACY OF THE ESTIMATED DISTRIBUTION. LARGEST VERTICAL ABSOLUTE DEVIATION BETWEEN ESTIMATED AND SAMPLE C.D.F. Minimum, median and the maximum largest absolute deviation in 100 simulations computed over all links as function of n (a) and link by link for $n = 10000$ (b).

edge (1Mb/sec and 10ms). Each link is modeled as a FIFO queue with a 4-packet capacity.

Node 0 generates probes as a 20Kbit/s stream comprising 40 byte UDP packets according to a Poisson process with a mean interarrival time of 16ms; this represents 2% of the smallest link capacity. Observe that even for this simple topology with 8 end-points, a mesh of unicast measurements with the same traffic characteristics would require an aggregate bandwidth of 160Kbit/s at the root. The background traffic comprises a mix of infinite data source TCP connections (FTP) and exponential on-off sources using UDP. Averaged over the different simulations, the link loss ranges between 1% and 11% and link utilization ranges between 20% and 60%.

For a single experiment, Figure 11 compares the estimated versus the sample average delay for representative selected links. The analysis has been carried out using $d = 1ms$ (a) and $d = 0.1ms$ (b). In this example, we practically obtain the same accuracy despite a tenfold difference in resolution. (Observe that $d = 1ms$ is of the same order of magnitude of the average delays.) The inferred averages rapidly converge to the sample averages even though we have persistent systematic errors in the inferred values due to consistent spatial correlation. We shall comment upon this later.

In order to show how the inferred values not only quickly converge, but also exhibit good dynamics tracking, in Figure 12 we plot the inferred versus the sample average delay for 3 links (1, 3 and 10) computed over a moving window of two different sizes with jumps of half its width. To allow greater dynamics, here we arranged background sources with random start and stop times. Under both window sizes (approximately 300 and 1200 probes are used, respectively), the esti-

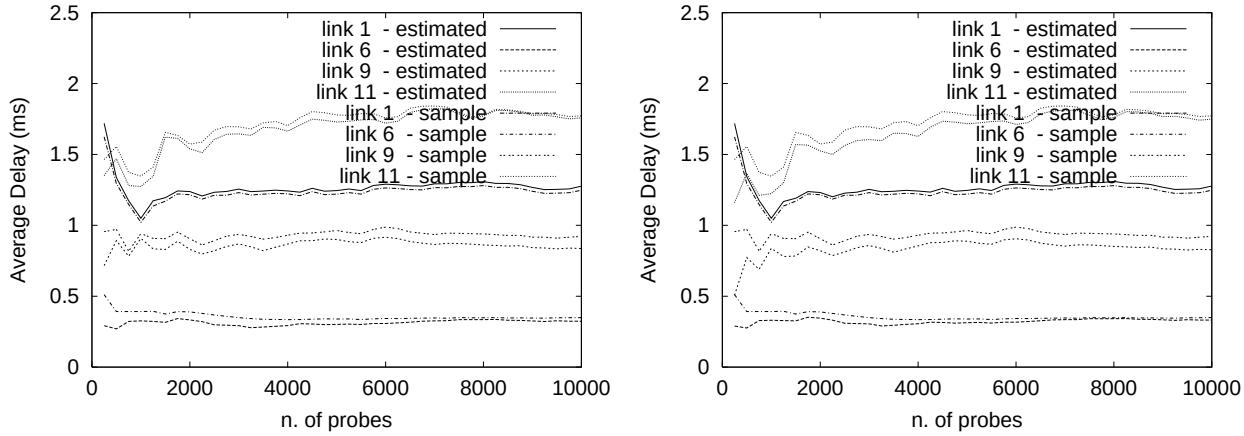


Figure 11: CONVERGENCE OF INFERRED VERSUS SAMPLE AVERAGE LINK DELAY IN TCP/UDP SIMULATIONS. (a): bin-size $d = 1ms$. (b): bin size $d = 0.1ms$. The graphs shows how the inferred values closely track the sample average delays.

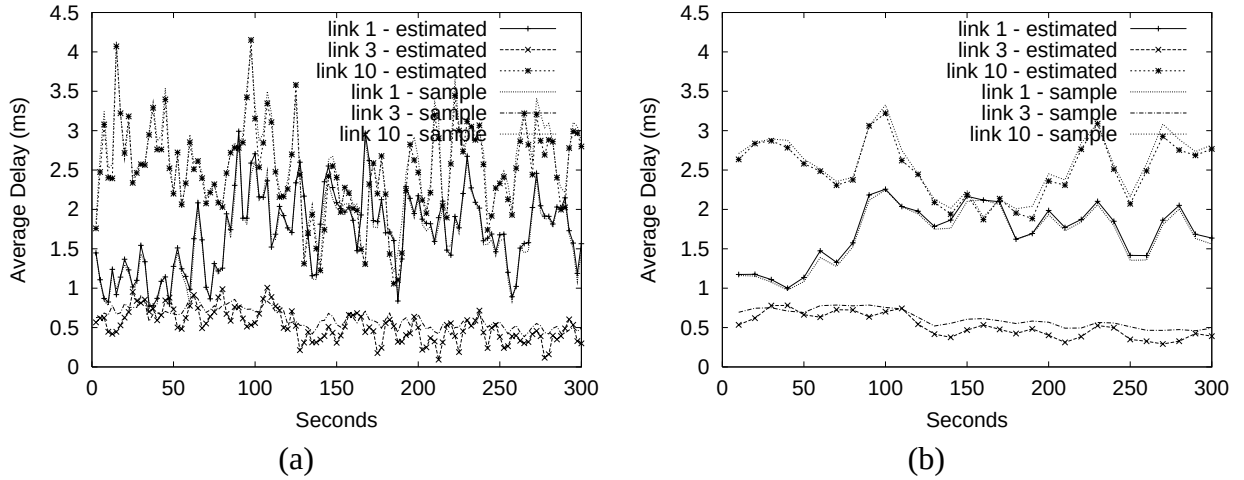


Figure 12: DYNAMIC ACCURACY OF INFERENCE. Sample and Inferred average delay on links 1, 3 and 10 of the multicast tree in Figure 8. (a): 5 seconds window. (b): 20 seconds windows. Background traffic has random start stop times.

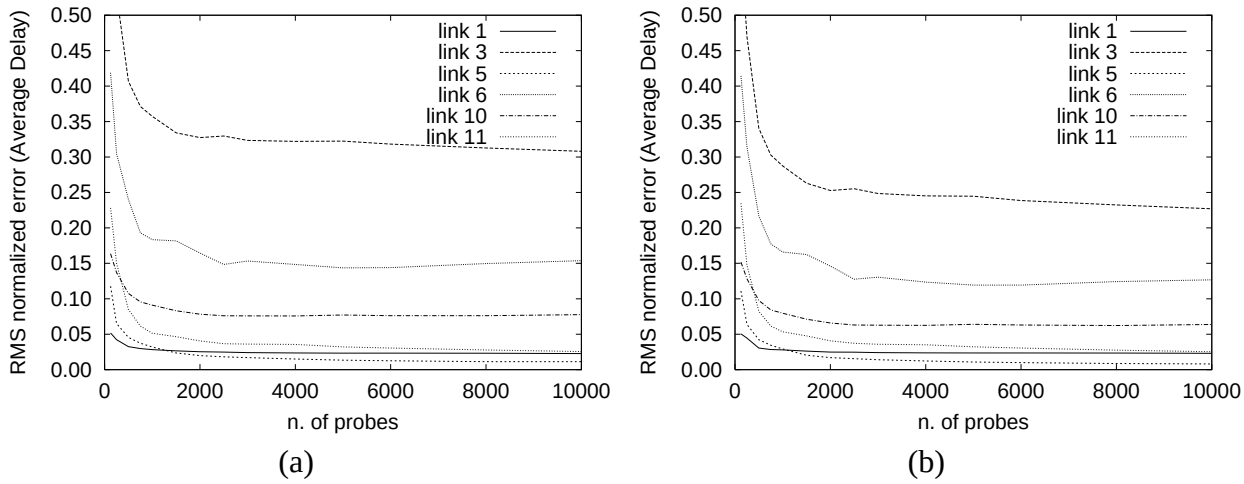


Figure 13: ACCURACY OF INFERENCE: AVERAGE DELAY. Left: $d = 1ms$. Right: $d = 0.1ms$. The graphs show the normalized Root Mean Square error between the estimated and sample average delay over 100 simulations.

mates of the average delays of links 1 and 10 show good agreement and a quick response to delay variability revealing a good convergence rate of the estimator. For link 3 with a smaller average delay, the behavior is rather poor, especially for the 5 seconds windows size.

For a selection of links, in Figure 13 we plot the Root Mean Square (RMS) normalized error between the estimated and sample average delays calculated over 100 simulations using $d = 1ms$ and $d = 0.1ms$. The two plots demonstrate that the error drops significantly up to 2000 probes after which it becomes almost constant. In this example, increasing the resolution by a factor of ten improves, although not significantly, the overall accuracy of the estimates especially for those links that enjoy smaller delays. After 10000 probes the relative error ranges from 1% to 23%. The higher values occur when link average delays are small due to the fact that for these links the same absolute error results in a more pronounced relative error.

The persistence of systematic errors we observe in Figure 13 is due to the presence of spatial correlation. In our simulations, a multicast probe is more likely to experience similar level of congestion on consecutive links or on sibling links than is dictated by the independence assumption. We also verified the presence of temporal correlation among successive probes on the same link caused by consecutive probes experiencing the same congestion level at a node.

To assess the extent to which our real traffic simulations violate the model assumptions, we computed the delay correlation between links pairs and among packets on the same link. The analysis revealed the presence of significant spatial correlations up to $0.3 \sim 0.4$ between consecutive links. The smallest values are observed for link 5 which always exhibits a correlation with its

parent node that lies below 0.1. From Figure 13 we verify that, not surprisingly node 5 enjoys the smallest relative error. We believe that these high correlations are a result of the small scale of the simulated network. We have observed smaller correlations in large simulations as would be expected in real networks because of the wide traffic and link diversity.

The autocorrelation function rapidly decreases and can be considered negligible for a lag larger than 30 (approximatively 2 seconds). The presence of short-term correlation does not alter the key property of convergence of the estimator as it suffices that the underlying processes be stationary and ergodic (this happens for example, when recurrence conditions are satisfied). The price of correlation, however, is that the convergence rate is slower than when delay are independent.

Now we turn our attention to the inferred distributions. For an experiment of 300 seconds during which approximately 18000 probes were generated, we plot the complementary c.d.f. conditioned on the delay being finite in Figures 14. In Figure 15 we also plot the complement c.d.f of the node cumulative delay. (we show only the internal links as $\widehat{A}_k(i) = \widehat{\gamma}_k(i), k \in R$). Here $d = 1ms$.

From these two sets of plots, it is striking to note the differences between the accuracy of the estimated cumulative delay distributions $\widehat{A}_k$ and the estimated link delay distributions $\widehat{\alpha}_k$: while the former are all very close to the actual distributions, the latter results are inaccurate in many cases. This is explained by observing that in presence of significant correlations, the convolution among A_k, α_k, and $A_{f(k)}$, used in the model, does not well capture the relationship between the actual distributions. We verified this by convolving α_k and $A_{f(k)}$ and comparing the result with A_k; as expected, in the presence of strong local correlation, the results exhibit significant differences that account for the discrepancies of the inferred distributions. Nevertheless, results should be affected in a continuous way with small violations leading to small inaccuracies. Indeed, we have good agreement for the inferred distributions of links 4, 5, 10 and 12 that are the nodes with smallest spatial correlations. Unfortunately it is not easy to determine whether the correlations are strong and therefore assess the expected accuracy of the estimates, even though pathological shapes of the inferred distributions could provide evidence of strong local correlations[1]. A solution could be the extension of the model to explicitly account for the presence of spatial correlation in the analysis. This will be the focus of future research.

The accuracy of the inferred cumulative delay distributions, on the other hand, derives from the fact that even in presence of significant local correlations, equation (8), which assumes inde-

[1]To this end, we observed that under strong spatial correlation inaccuracies of the estimator $\widehat{\alpha}$ are often associated to the existence of significant increasing behavior portions in the complement c.d.f. that reveals the presence of negative inferred probabilities with possibly non negligible absolute values.

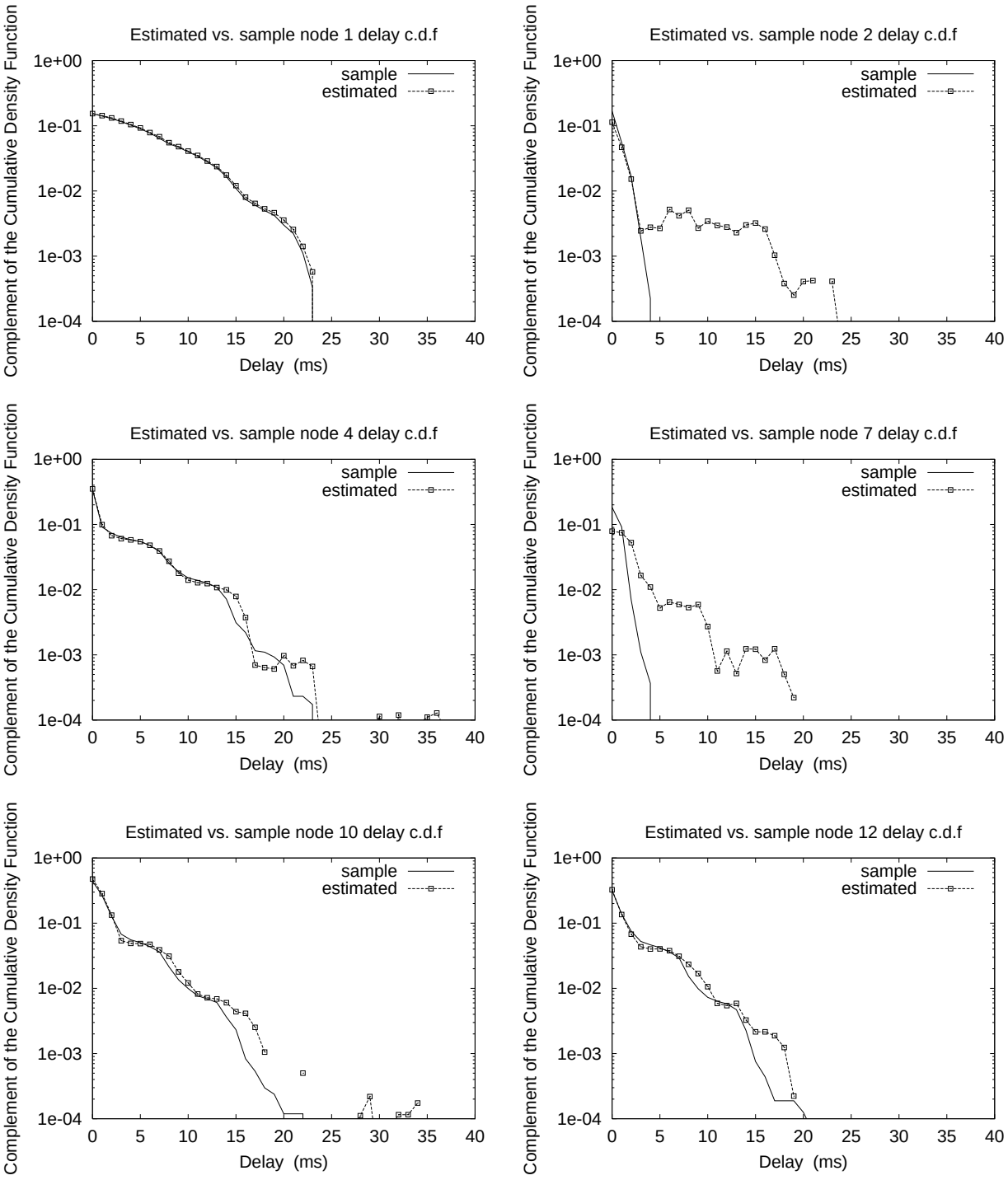


Figure 14: Sample vs. Estimated Delay c.d.f. for selected links.

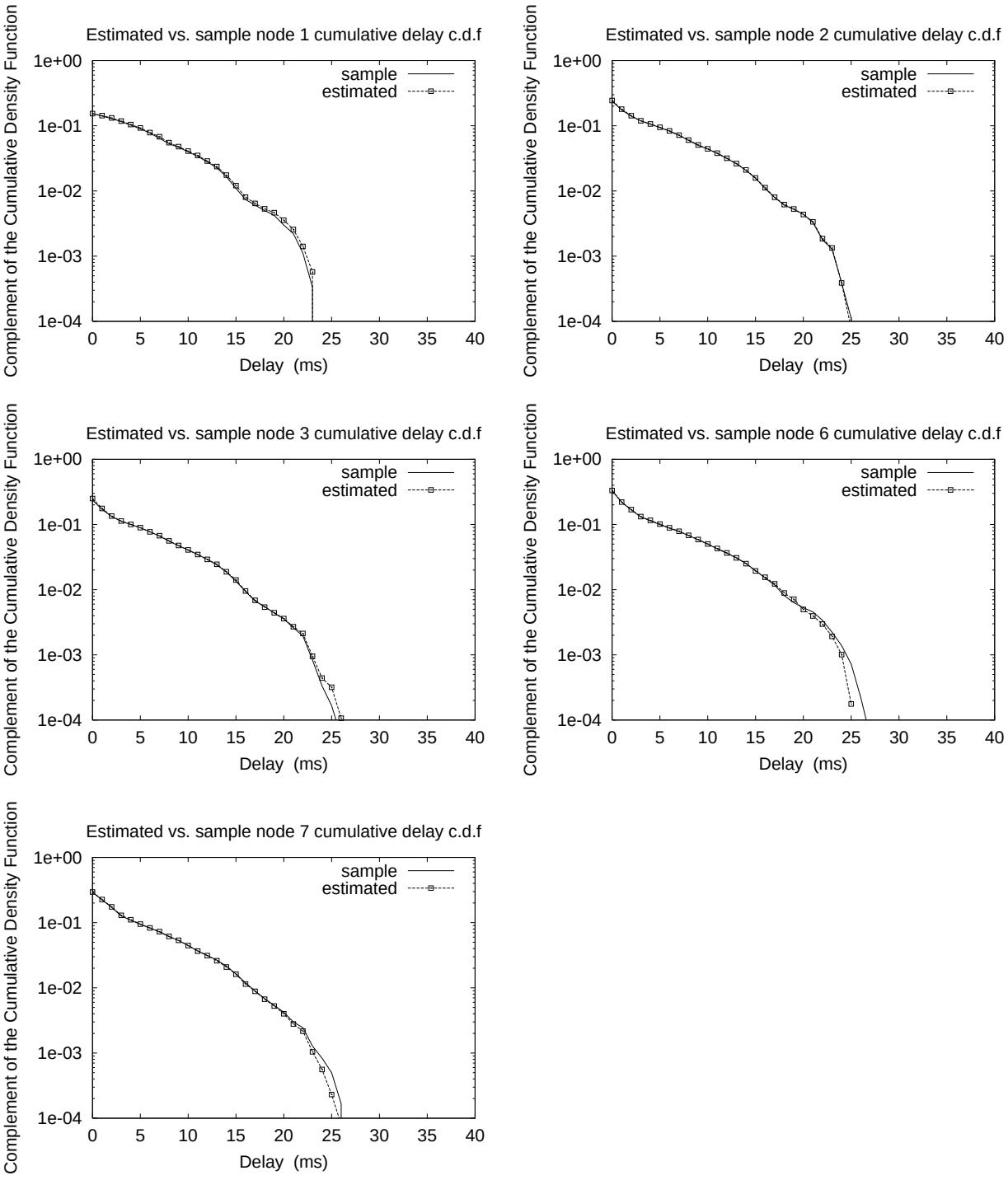


Figure 15: Sample vs. Estimated node k cumulative delay c.d.f.

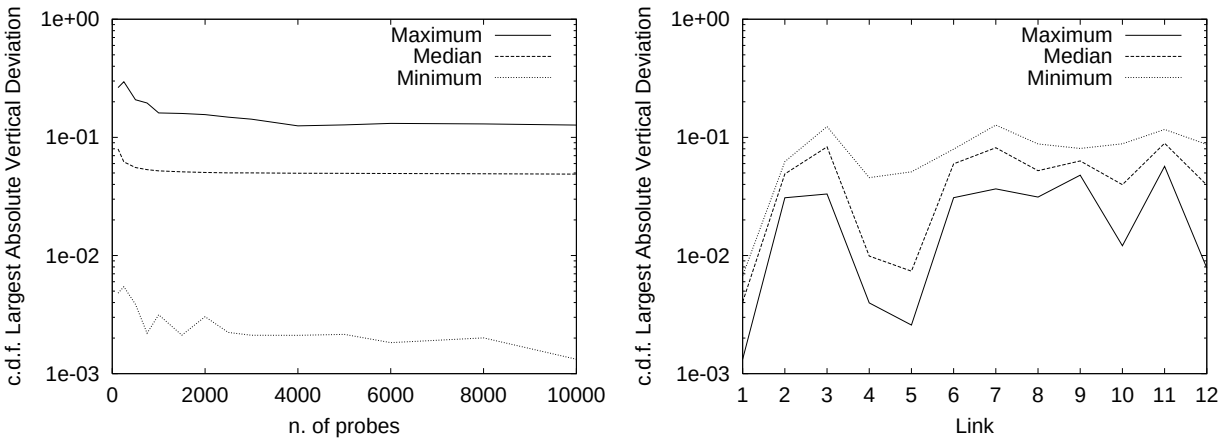


Figure 16: TCP/UDP SIMULATION: TOPOLOGY OF FIGURE 8. ACCURACY OF THE ESTIMATED DISTRIBUTION. LARGEST VERTICAL ABSOLUTE DEVIATION BETWEEN ESTIMATED AND THEORETICAL C.D.F. Minimum, median and the maximum largest absolute deviation in 100 simulations computed over all links as function of n (a) and link by link for $n = 10000$ (b).

pendence, is still accurate. This can be explained by observing that (8) is equivalent to (4) which consists of a convolution between $A_{f(k)}$ and β_k; we expect the correlation between the delay accrued by a probe in reaching node $f(k)$ and the minimum delay accrued from node $f(k)$ to reach any receiver be rather small, especially as the tree size grows, as these delays span the entire multicast tree.

Finally in Figure 16 we plotted the minimum, median and maximum largest deviation between inferred and theoretical c.d.f. over 100 simulations computed over all links as function of n (left) and link by link as for $n = 10000$ (right). Due to spatial correlation, the largest deviation level off after the first 2000 probes with the median that stabilize at 5%. The accuracy again exhibits a negative trend as we descend the tree.

6 Conclusions and Future Work

In this paper, we introduced the use of end-to-end multicast measurements to infer network internal delay in a logical multicast tree. Under the assumption of delay independence, we derived an algorithm to estimate the per link discrete delay distributions and utilization from the measured end-to-end delay distributions. We investigated the statistical properties of the estimator, and show it to be strongly consistent and asymptotically normal.

We evaluated our estimator through simulation. Within model simulation we verified the accuracy and convergence of the inferred to the actual values as predicted by our analysis. In real

traffic simulations, we found rapid convergence, although some persistent difference to the actual distributions because of spatial correlation.

We are extending our delay distribution analysis in several directions. First we plan to do more extensive simulations, exploring larger topologies, different node behavior, background traffic and probe characteristics. Moreover, we are exploring how probe delay is representative of the delay suffered by other applications and protocols, for examples TCP.

Second, we are analyzing the effect of spatial correlation among delays and we are planning to extend the model by directly taking into account the presence of correlation. Moreover, we are studying the effect of the choice of the bin size on the accuracy of the results. To deal with continuously distributed delay, we derived a continuous version of the inference algorithm we are currently investigating.

Finally, we believe that our inference technique can shed light on the behavior and dynamics of per link delay and so allow us to develop accurate link delay models. This will be also object of future works.

We feel that multicast based delay inference is an effective approach to perform delay measurements. The techniques developed are based on rigorous statistical analysis and, as our results show, yield representative delay estimates for all traffic which receive the same per node behavior of multicast probes. The approach does not depend on cooperation from networks elements and because of bandwidth efficiency of multicast traffic is well suited to cope with the growing size of today networks.

References

[1] R. Bellmann and R. Roth, "The Laplace Transform", World Scientific, Singapore 1984.

[2] J. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet." *Journal of High-Speed Network*, vol. 2 n. 3, pp. 289-298, Dec. 1993.

[3] J-C. Bolot and A. Vega Garcia "The case for FEC-based error control for packet audio in the Internet" *to appear in ACM Multimedia Systems*.

[4] R. Caceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics" *to appear in IEEE Trans. on Information Theory, November 1999*.

[5] R. Caceres, N.G. Duffield, J .Horowitz, D. Towsley and T. Bu, "Multicast-Based Inference of Network Internal Loss Characteristics: Accuracy of Packet Estimation" *Proc. of Infocom '99*, New York, NY, Mar. 1999.

[6] R. Caceres, N.G. Duffield, S. Moon, and D. Towsley, "Inferring Link-Level Performance from End-to-End Measurements" *submitted for publication*, Mar. 1999.

[7] R. Caceres, N.G. Duffield, J .Horowitz, F. Lo Presti and D. Towsley, "Loss-Based Inference of Multicast Network Topology" *to appear in Proc. of 1999 IEEE Conference on Decision and Control*, Phoenix, AZ, Dec. 1999.

[8] K. Claffy, G. Polyzos and H-W. Braun, "Measurements Considerations for Assessing Unidirectional Latencies", *Internetworking: Research and Experience*, Vol. 4, no. 3, pp. 121-132, Sept. 1993.

[9] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE '96*, Oct. 1996.

[10] A. Downey, "Using `pathchar` to estimate Internet link characteristics, *Proc. SIGCOMM 1999*, Cambridge, MA, pp. 241-250, Sept. 1999.

[11] N. Duffield, F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Networks Links" *submitted to IEEE Infocom 2000*.

[12] N. Duffield, J. Horowitz, F. Lo Presti and D. Towsley, "Probabilistic Inference Methods for Multicast Network Topology", *in preparation*.

[13] Felix: Independent Monitoring for Network Survivability. For more information see `ftp://ftp.bellcore.com/pub/mwg/felix/index.html`

[14] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, no. 4, August 1993.

[15] IPMA: Internet Performance Measurement and Analysis. For more information see `http://www.merit.edu/ipma`

[16] IP Performance Metrics Working Group. For more information see `http://www.ietf.org/html.charters/ippm-charter.html`

[17] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see `ftp://ftp.ee.lbl.gov/pathchar`

[18] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, "Creating a Scalable Architecture for Internet Measurement," *Proc. of INET '98*, Geneva, Switzerland, July 1998.

[19] M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, Montreal, June 1996.

[20] D. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", *RFC 1305*, Network Information Center, SRI International, Menlo Park, CA, Mar. 1992.

[21] S. Moon, P. Skelly and D. Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements" *Proc. of Infocom '99*, New York, NY, Mar. 1999.

[22] S. Moon, J. Kurose, P. Skelly and D. Towsley, "Correlation of Packet Delay and Loss in the Internet" *Tech. Report University of Massachussets at Amherst*, 1999.

[23] `mtrace` – Print multicast path from a source to a receiver. For more information see `ftp://ftp.parc.xerox.com/pub/net-research/ipmulti`

[24] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load", *Internetworking: Research and Experience*, Vol. 5, pp. 163-205, Dec. 1994.

[25] `ns` – Network Simulator. For more information see `http://www-mash.cs.berkeley.edu/ns/ns.html`

[26] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. SIGCOMM '96*, Stanford, Aug. 1996.

[27] V. Paxson, "End-to-End Internet Packet Dynamics," *Proc. SIGCOMM 1997*, Cannes, France, pp. 139-152, Sept. 1997.

[28] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. Dissertation, University of California, Berkeley, Apr. 1997.

[29] V. Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. SIGCOMM 1997*, Cannes, France, pp. 167-179, Sept. 1997.

[30] V. Paxson, "On calibrating measurements of Packet Transit Times", *Proc. of SIGMETRICS '98*, Madison, June 1998.

[31] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", Proceedings IEEE Infocom' 99, New York, NY, Mar. 1999.

[32] D. Sanghi, A. Agrawala and B. Jain, "Experimental assessment of end-to-end behavior on Internet", *Proc. IEEE Infocom '93*, San Francisco, CA, pp. 867-874, Mar. 1993.

[33] D. Sanghi, O. Gudnumdsson, A. K. Agrawala, "Study of network dynamics", *Proc. 4th Joint European Networking Conference*, Trondheim, Norway, pp. 241-249, May 1993.

[34] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[35] Surveyor. For more information see `http://io.advanced.org/surveyor/`

A Uniqueness and Continuously Differentiability of the Inverse

The algorithm presented in Section 3.2.2 computes a solution of the system of equations (5) in the unknown $A = (A_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$ given $\gamma = (\gamma_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$. By deconvolution we then compute $\alpha = (\alpha_k(i))_{k\in V, i\in\{0,\ldots,i_{\max}\}}$.

We now show that solution so computed is the unique solution of the equation $\gamma = \Gamma(\alpha)$, i.e. that it is uniquely defined the inverse $\alpha = \Gamma^{-1}(\gamma)$. To this end we rewrite the mapping $\alpha = \Gamma(\gamma)$ as $\gamma = \chi \circ \psi(\alpha)$ where $A = \psi(\alpha)$ is clearly a bijection. It remains to show that χ is also a bijection. To prove this consider α, A and γ such that $A = \psi(\alpha)$ and $\gamma = \chi(A)$. We first show that $A_k(i)$, $i = 1, \ldots, i_{\max}$ is the second largest solution of (8).

In the binary case we can directly solve (8) to obtain the two solutions

$$A_k(i) + A_k(0)\left(\frac{\beta_{d_1}(i)}{\beta_{d_1}(0)} + \frac{\beta_{d_2}(i)}{\beta_{d_2}(0)} - 1\right) \geq A_k(i) + A_k(0)$$

and

$$A_k(i)$$

For the general case we have the following Lemma

Lemma 1 *Let $x_1 \geq x_2 \geq \ldots \geq x_m$, $m \leq \#d(k)$ denote the real solutions of the equation*

$$\begin{array}{c}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1 - \frac{\gamma_d(i) - \sum_{j=1}^{i-1}\beta_d(i-j)A_k(j) - \beta_d(0)x}{A_k(0)}\right] - 1\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + x\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.\end{array} \tag{16}$$

Then $x_2 = A_k(i)$.

Proof: Substitute $x = A_k(i) + yA_k(0)$ in equation (16) obtaining

$$\begin{array}{c}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1 - \beta_d(i) + \beta_d(0)y\right] - 1\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + (A_k(i) + yA_k(0))\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.\end{array} \tag{17}$$

To prove the lemma we simply need to show that $y = 0$ is the second largest solution of (17). Expanding the product in the second term we get

$$\begin{array}{c}\gamma_k(i) + A_k(0)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i)\right] - 1\right\} + A_k(0)\left\{\sum_{b\in B}\prod_{m\in\{1,\ldots,\#d(k)\}}(1-\beta_{d_m}(i))^{b_m}\beta_{d_m}(0)^{1-b_m}y^{1-b_m}\right\} + \\ \sum_{j=1}^{i-1} A_k(j)\left\{\prod_{d\in d(k)}\left[1-\beta_d(i-j)\right] - 1\right\} + (A_k(i) + yA_k(0))\left\{\prod_{d\in d(k)}\left[1-\beta_d(0)\right] - 1\right\} = 0.\end{array}$$

where $b = \{b_1, \ldots, b_{\#d(k)}\}$, $B = \{0,1\}^{\#d(k)} \setminus \{0\}^{\#d(k)}$. Observing that the constant terms sums to 0 (equation (5) and dividing by $A_k(0)$) this reduces to

$$\sum_{b\in B} \prod_{m\in\{1,\ldots,\#d(k)\}} (1-\beta_{d_m}(i))^{b_m} \beta_{d_m}(0)^{1-b_m} y^{1-b_m} + y\left\{\prod_{d\in d(k)}[1-\beta_d(0)] - 1\right\} = 0.$$

Grouping with respect to y^l, we obtain

$$\tilde{\theta}_{k,i}(y) = \sum_{l=1}^{\#d(k)} y^l \sum_{b\in B, \sum b_m = \#d(k)-l} \prod_{m\in\{1,\ldots,\#d(k)\}} (1-\beta_{d_m}(i))^{b_m} \beta_{d_m}(0)^{1-b_m} + y\left\{\prod_{d\in d(k)}[1-\beta_d(0)] - 1\right\} = 0.$$

The coefficients of the polynomial are all positive but the last which is negative. The proof follows observing that since $\tilde{\theta}_{k,i}(0) = 0$, $\tilde{\theta}'_{k,i}(0) < 0$ and $\tilde{\theta}''_{k,i}(y) > 0$, $y \geq 0$, there is one and only one solution of (17) greater than zero. ∎

From the uniqueness of the solution of (6) for canonical delay trees and by induction on i, it then follows that χ is a bijection; thus, the inverse is uniquely defined.

To prove that the inverse is continuously differentiable we proceed as follows. Denote $\theta_{k,i}(\gamma, A)$, $k \in U$, $i = 0, \ldots, i_{\max}$, the left hand side of (8). Define the function $H(\gamma, A) = (\theta_{k,j}(\gamma, A))_{k\in U, j=0,\ldots,i_{\max}}$. $H(\gamma, A) = 0$ is the system of equations to be solved to compute A given γ. Denote $A(\gamma)$ the unique solution to $H(\gamma, A(\gamma)) = 0$. The proof that the inverse is continuously differentiable amounts to show that so is $A = \chi^{-1}(\gamma) = A(\gamma)$ (as ψ and its inverse clearly are). For canonical trees, $A_k(0) > 0$, $k \in U$, and therefore H is continuously differentiable. Then, by the Implicit Function Theorem, so is $A(\gamma)$ provided that the determinant of the Jacobian $\frac{\partial H}{\partial A}\Big|_{A=A(\gamma)}$ is different from zero. To see this, observe that $\frac{\partial \theta_{k_1,i_1}(\gamma,A)}{\partial A_{k_2,i_2}} = 0$ if $k_1 \neq k_2$ or $i_1 < i_2$; hence the Jacobian matrix is always triangular. The diagonal elements are $\frac{\partial \theta_{k,i}(\gamma,A)}{\partial A_{k,i}}\Big|_{A=A(\gamma)} = \tilde{\theta}'_{k,i}(0)/A_k(0) < 0$.

B The Continuous Model: Delay Distribution Analysis

In this Appendix we formulate the delay analysis for continuous delay distributions, rather than for the discrete distributions. We assume now that $D_k \in \mathbb{R}_+ \cup \{\infty\}$ and the distribution to be absolutely continuous w.r.t Lebesgue measure on $\mathbb{R}_+$ with density α_k, together with an atom at ∞ of mass $\alpha_k(\infty) = 1 - \int_0^\infty \alpha_k(x)dx$, the probability that a packet is lost traversing the link terminating at k. (For simplicity of notation, here we do not consider the atom in 0 representing the probability that a probe experiences minimum delay.) A_k is defined similarly for the source to root delays Y_k. Similar to the discrete case we define $\Omega_k(x)$ as the event $\{\min_{j\in R(k)} Y_j \leq x\}$ and $\gamma_k(x) = \mathsf{P}[\Omega_k(x)]$, $k \in U$. Finally, let $\beta_k(x) = \mathsf{P}[\min_{j\in R(k)} Y_j - Y_{f(k)} \leq x]$, $k \in U$. From the

above definitions, the following relations hold:

$$A_k(x) = \int_0^x \alpha_k(y) A_{f(k)}(x-y)dy,\ k \in U,$$

where we set $A_0(x) = \delta(x)$,

$$\beta_k(x) \quad = \quad \int_0^x \alpha_k(y)(1 - \prod_{d \in d(k)} [1 - \beta_d(x-y)])dy,\ k \in U \tag{18}$$

and

$$\gamma_k(x) \quad = \quad \int_0^x A_k(y)(1 - \prod_{d \in d(k)} [1 - \beta_d(x-y)])dy,\ k \in U. \tag{19}$$

which is the continuous version of (5). Empty products are assumed to be equal to zero.

The above equation can be rewritten in more convenient form using the Laplace transform

$$\gamma_k(s) = A_k(s) \left[\frac{1}{s} - *_{d \in d(k)} \left(\frac{1}{s} - \frac{\gamma_d(s)}{A_k(s)} \right) \right]\ k \in U \tag{20}$$

or

$$\frac{1}{s} - \frac{\gamma_k(s)}{A_k(s)} = *_{d \in d(k)} \left[\frac{1}{s} - \frac{\gamma_d(s)}{A_k(s)} \right],\ k \in U \tag{21}$$

where, $f(s) = \int_0^\infty f(x) e^{-sx} dx$ is the Laplace transform of $f(x)$, $*$ is the convolution operator in the domain s, $f(s) * g(s) = \int_{a-i\infty}^{a+i\infty} f(p) g(s-p) dp$, and $\beta_d(s) = \frac{\gamma_d(s)}{A_k(s)}$.

Given $\gamma_k(s)$, $k \in U$, (21) represents a system of $\#U$ independent equations in the unknown $A_k(s)$, $k \in U$. $\alpha_k(s)$ can then be computed by the quotients

$$\alpha_k(s) = \frac{A_k(s)}{A_{f(k)}(s)},\ k \in U.$$

Solving equation (21) is not trivial especially because of the convolution in the right hand side which in general can be computed only numerically. Furthermore, we have not been able yet to establish whether the solution is unique. The inversion of the Laplace transform poses other challenges. It is well known, indeed (see for example [1]), that the Laplace inverse transform is an unbounded operator. In other words arbitrary small changes in the transform will produce arbitrary large changes in the original function. Therefore it may not be easy to control the accuracy of the results obtained with such an approach. All these issues will be subject of further investigation. The estimator $\widehat{\alpha}(s)$ can be computed in the same manner from the estimates $\widehat{\gamma}(s)$ we obtain from the measurements. (21) can be written as a fixed point equation for the $A_k(s)$; this suggests the possible use of contraction mapping theorem in order to establish existence and uniqueness of solutions.

C Proof of Theorem 4

The proof proceeds by a number of subsidiary results.

C.1 Limit Behavior of A, β and γ

As $\| \alpha \| \to 0$,

(i)

$$A_k(i) = \begin{cases} 1 - s_k(0) + O(\| \alpha \|^2) & i = 0 \\ \sum_{l=0}^{\ell(k)} \alpha_{f^l(k)}(i) + O(\| \alpha \|^2) & i > 0 \end{cases} \tag{22}$$

(ii)

$$1 - \beta_k(i) = \sum_{j>i} \alpha_k(i) + O(\| \alpha \|^2) \tag{23}$$

(iii)

$$\gamma_k(i) = 1 - s_k(i) + O(\| \alpha \|^2) \tag{24}$$

where

$$s_k(i) = \sum_{l=0}^{\ell(k)} \sum_{j>i} \alpha_{f^l(k)}(j) \tag{25}$$

The relation (i) is clear for $i = 0$ by expanding $A_k(0) = \prod_{l=0}^{\ell(k)} \alpha_k(0)$; for $i > 0$, it follows by an inductive argument on k and i: it is true for $k = 1$ and $i = 1$; if it is true for $j \succ k$ and for $0 \leq i' \leq i - 1$, then

$$A_k(i) = \sum_{j=0}^{i} A_{f(k)}(j)\alpha_k(i-j) = A_{f(k)}(0)\alpha_k(i) + A_{f(k)}(l)\alpha_k(0) + O(\| \alpha \|^2) = \alpha_k(i) + \sum_{l=0}^{\ell(f(k))} \alpha_{f^l(k)}(i) + O(\| \alpha \|^2).$$

Also (ii) follows by an inductive argument. Observe from (3) that if (ii) hold for all j in $d(k)$ and $i' \leq i$ then $1 - \beta_k(i) = 1 - \sum_{j=0}^{i} \alpha_k(j) + O(\| \alpha \|^2)$. Since $\beta_k(i) = \sum_{j=0}^{i} \alpha_k(j)$ for $k \in R$, $i = 1, \ldots, i_{\max}$, (ii) holds for all k and i. (iii) then follows expanding $\gamma_k(i) = \sum_{j=0}^{i} A_k(j)[1 - \prod_{d \in d(k)} \beta_d(i-j)]$ (Observe that the terms within square bracket are always of the form $1 - O(\| \alpha \|)$).

C.2 Limit Behavior of the Covariance Matrix σ

As $\| \alpha \| \to 0$,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1 \wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2) \tag{26}$$

where $k_1 \wedge k_2$ denote the minimal common ancestor of k_1 and k_2 with respect to $\prec$.

To see this, we write $\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = \mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] - \mathsf{E}[\widehat{\gamma}_{k_1}(i_1)]\mathsf{E}[\widehat{\gamma}_{k_2}(i_2)]$. By definition, $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)] = \gamma_{k_1}(i_1)$ and $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] = \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cap \min_{m\in R(k_2)} Y_m \leq i_2 q]$ [2]. We have three cases:

(i) $k_1 \succeq k_2$**,** $i_1 \geq i_2$**.**

In this case $\{\min_{m\in R(k_2)} Y_m \leq i_2 q\} \subseteq \{\min_{m\in R(k_1)} Y_m \leq i_1 q\}$ and $\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)\widehat{\gamma}_{k_2}(i_2)] = \widehat{\gamma}_{k_2}(i_2)$. Thus,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = (1 - \gamma_{k_1}(i_1))\gamma_{k_2}(i_2) = s_{k_1}(i_1) + O(\| \alpha \|^2).$$

(26) follows as $i_1 = \max(i_1, i_2)$ and $k_1 = k_1 \wedge k_2$.

(ii) $k_1 \succ k_2$**,** $i_1 < i_2$**.**

Write $\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cap \min_{m\in R(k_2)} Y_m \leq i_2 q] = \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q] + \mathsf{P}[\min_{m\in R(k_2)} Y_m \leq i_2 q] - \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q]$. The first two terms are $\gamma_{k_1}(i_1)$ and $\gamma_{k_2}(i_2)$. Then, as for $\| \alpha \| \to 0$

$$\mathsf{E}[\widehat{\gamma}_{k_1}(i_1)]\mathsf{E}[\widehat{\gamma}_{k_2}(i_2)] = 1 - s_{k_1}(i_1) - s_{k_2}(i_2) + O(\| \alpha \|^2), \tag{27}$$

it readily follows that as $\| \alpha \| \to 0$

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = 1 - \mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] + O(\| \alpha \|^2). \tag{28}$$

To compute $\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q]$ we need to define some additional quantities. Denote $W = \{w_1, \ldots, w_l\} \subset V$ a set of nodes that induces a partition on R, *i.e.*, W is such that $\cup_{h=1}^{l} R(w_h) = R$ and $R(w_h) \cap R(w_{h'}) = \emptyset$, $1 \leq h, h' \leq l$, $h \neq h'$. Associate to W a set of delay values $\Delta = \{j_1, \ldots, j_l\}$. The quantities we introduce below are a generalization of the β and γ, where we use for different sets of receivers, namely $R(w_1), \ldots, R(w_l)$, different delays, $j_1, \ldots, j_l$. For $k \in V$, define

$$\chi_{k,W}(\Delta) = \mathsf{P}[\cup_{h=1}^{l} \min_{w\in R(w_h)\cap R(k)} Y_m - Y_{f(k)} \leq i_h q]. \tag{29}$$

Then, $\chi_{k,W}$ obey to the recursion

$$\chi_{k,W}(\Delta) = \sum_{j=0}^{\max_{j_{k,W}}} \alpha_k(j) \left[1 - \prod_{d\in d(k)} 1 - \chi_{d,W}(\Delta - j)\right], \; k \in U \setminus R \tag{30}$$

$$\chi_{k,W}(\Delta) = \beta_k(j_{k,W}), \; k \in R. \tag{31}$$

[2] Since probes are assumed independent, it suffices to evaluate all random quantities for $n = 1$ probes.

where $\Delta - j = \{j_1 - j, \ldots, j_l - j\}$ and $j_{k,W} = \max_{l':R(k)\cap R(w_{l'})\neq\emptyset} j_{l'}$. Probabilities with negative index are assumed to be equal to zero.

For a given node k, define now

$$\eta_{k,W}(\Delta) = \mathsf{P}[\cup_{h=1}^{l} \min_{w\in R(w_h)\cap R(k)} Y_m \leq j_h q]. \tag{32}$$

The following, which can be regarded as a generalization of (5), holds

$$\eta_{k,W}(\Delta) = \sum_{j=0}^{j_{k,W}} A_k(j) \left[1 - \prod_{d\in d(k)} 1 - \chi_{d,W}(\Delta - j)\right], \; k \in U \setminus R \tag{33}$$

$$\eta_{k,W}(\Delta) = \sum_{j=0}^{j_{k,W}} A_k(j), \; k \in R. \tag{34}$$

For $\| \alpha \| \to 0$, it is easy to verify that $\eta_{k,W}(\Delta)$ behaves as $\gamma_k(j_{k,W})$ (the terms within square bracket are always of the form $1 - O(\| \alpha \|)$). In other words, $\eta_{k,W}(\Delta) = 1 - s_k(j_{k,W}) + O(\| \alpha \|^2)$.

With the definitions above, we can now write

$$\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] = \eta_{k_1,W}(\Delta) \tag{35}$$

where $W = \{w_1, \ldots, w_l\}$, $w_1 = k_2$, and $\Delta = \{j_1, \ldots, j_l\}$, with $j_1 = i_2$ and $j_{l'} = i_1$, $2 \leq l' \leq l$. Then, as $(\| \alpha \|) \to \infty$,

$$\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] = 1 - s_{k_1}(i_2) + O(\| \alpha \|^2). \tag{36}$$

Thus,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1}(i_2) + O(\| \alpha \|^2). \tag{37}$$

(26) follows as $k_1 = k_1 \wedge k_2$ and $i_2 = \max(i_1, i_2)$.

(iii) $k_1 \wedge k_2 \succ k_1, k_2$**.**

We proceed as for **(ii)**. In this case, we can write

$$\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] = \eta_{k_1\wedge k_2,W}(\Delta) \tag{38}$$

where $W = \{w_1, \ldots, w_l\}$, $w_1 = k_1$ and $w_2 = k_2$, and $\Delta = \{j_1, \ldots, j_l\}$, with $j_1 = i_1$, $j_2 = i_2$ and $j_{l'} = -1$, $3 \leq l' \leq l$ (for (38) to hold, we need to set j_l to -1, $l \neq 1, 2$, or any other negative number, to insure that all events regarding receivers different from $R(k_1)$ and $R(k_2)$ have probability zero). Thus, as $\| \alpha \| \to \infty$,

$$\mathsf{P}[\min_{m\in R(k_1)} Y_m \leq i_1 q \cup \min_{m\in R(k_2)} Y_m \leq i_2 q] = 1 - s_{k_1\wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2). \tag{39}$$

Thus,

$$\mathsf{Cov}(\widehat{\gamma}_{k_1}(i_1), \widehat{\gamma}_{k_2}(i_2)) = s_{k_1 \wedge k_2}(\max(i_1, i_2)) + O(\| \alpha \|^2). \tag{40}$$

C.3 Limit Behavior of the Jacobian $D(\alpha)$

As $\| \alpha \| \to 0$,

$$D(\alpha) = B \otimes D + O(\| \alpha \|) \text{ where } D_{k_1 k_2} = \begin{cases} 1 & k_2 = k_1 \\ -1 & k_2 = f(k_1) \\ 0 & \text{otherwise} \end{cases}, \tag{41}$$

where B is a $i_{\max} + 2 \times i_{\max} + 2$ matrix with entries $B_{ii'} = \delta_{ii'} - \delta_{ii'+1}$, $\otimes$ denote the Knonecker product and $\delta_{ii'} = 1$ if $i = i'$ and 0 otherwise.

To establish this, we first show that its inverse $D^{-1}(\alpha)$ whose elements are $(D^{-1}(\alpha))_{(k_1,i_1)(k_2,i_2)} = \partial \gamma_{k_1}(i_1) / \partial \alpha_{k_2}(i_2)$ has the following form for $\| \alpha \| \to 0$,

$$D^{-1}(\alpha) = L \otimes \tilde{D} + O(\| \alpha \|) \text{ where } \tilde{D}_{k_1 k_2} = \begin{cases} 1 & k_1 \preceq k_2 \\ 0 & \text{otherwise} \end{cases}, \tag{42}$$

where L is a unit lower triangular matrix, *i.e.*, $L_{ii'} = \mathbf{1}_{\{i \leq i'\}}$. To this end we rewrite $\gamma_k(i)$ as $\sum_{j=0}^{i} A_k(j)[1 - \prod_{d \in d(k)}(1 - \beta_d(i-j))]$. We have the following three cases:

(i) $k_1 \preceq k_2$**,** $i_1 \geq i_2$**.**

Let l' be such that $f^{l'}(k_1) = k_2$. Then, for $\| \alpha \| \to 0$

$$\begin{aligned}
\frac{\partial \gamma_{k_1}(i_1)}{\partial \alpha_{k_2}(i_2)} &= \sum_{j=i_2}^{i_1} \frac{\partial A_{k_1}(j)}{\alpha_{k_2}(i_2)} [1 - \prod_{d \in d(k_1)} (1 - \beta_d(i_1 - j))] && (43) \\
&= \sum_{j=i_2}^{i_1} \frac{\partial}{\alpha_{k_2}} \left(\sum_{\sum_{l=0}^{\ell(k_1)} j_l = j} \prod_{l=0}^{\ell(k_1)} \alpha_{f^l(k_1)}(j_l) \right) [1 - \prod_{d \in d(k_1)} (1 - \beta_d(i_1 - j))] && (44) \\
&= \prod_{l=0, l \neq l'}^{\ell(k_1)} \alpha_{f^l(k_1)}(0) [1 - \prod_{d \in d(k_1)} (1 - \beta_d(i_1 - i_2))] + \\
&\quad \sum_{j=i_2+1}^{i_1} \sum_{\sum_{l=0}^{\ell(k_1)} j_l = j, j_{l'} = i_2} \prod_{l=0, l \neq l'}^{\ell(k_1)} \alpha_{f^l(k_1)}(j_l) [1 - \prod_{d \in d(k_1)} (1 - \beta_d(i_1 - j))] && (45) \\
&= 1 + O(\| \alpha \|) && (46)
\end{aligned}$$

as the first term of (45) goes to 1 while the second goes to 0 because in any product there is at least one l such that $j_l > 0$.

(ii) $k_1 \succ k_2$**,** $i_1 \geq i_2$**.**

Let d' be such that there exists an l such that $f^l(k_2) = d'$. Then,

$$\begin{aligned}
\frac{\partial \gamma_{k_1}(i_1)}{\partial \alpha_{k_2}(i_2)} &= \sum_{j=0}^{i_1-i_2} A_{k_1}(j) \frac{\partial [1 - \prod_{d \in d(k_1)} (1 - \beta_d(i_1 - j))]}{\alpha_{k_2}(i_2)} && (47)\\
&= \sum_{j=0}^{i_1-i_2} A_{k_1}(j) \frac{\partial \beta_{d'}(i_1 - j)}{\alpha_{k_2}(i_2)} [1 - \prod_{d \in d(k_1), d \neq d'} (1 - \beta_d(i_1 - j))] && (48)\\
&= O(\| \alpha \|) && (49)
\end{aligned}$$

as each product goes to 0, as $\| \alpha \| \to 0$.

(iii) $i_1 < i_2$**.**

In this last case, $\gamma_{k_1}(i_1)$ does not depend on $\alpha_{k_2}(i_2)$ and the derivative is 0.

Since matrix inversion is continuous in an open neighborhood on non-singular matrices, then (41) follows since D and $\tilde{D}$ are inverses (see Section 10 of [4]) as also are L and B (trivial) and since for invertible square matrices F and G, $(F \otimes G)^{-1} = F^{-1} \otimes G^{-1}$.

C.4 Proof

From Theorem 3, (26), (41) and continuity of finite dimensional matrix products, we have for $\| \alpha \| \to 0$ that

$$\nu_{(k_1,i_1)(k_2,i_2)} = \sum_{k'_1,k'_2,i'_1,i'_2} D_{(k_1,i_1)(k'_1,i'_1)} s_{k'_1,k'_2}(\max(i'_1, i'_2)) D_{(k_2,i_2)(k'_2,i'_2)} + O(\| \alpha \|^2). \qquad (50)$$

It remains to evaluate

$$\begin{aligned}
\sum_{k'_1,k'_2,i'_1,i'_2} D_{(k_1,i_1)(k'_1,i'_1)} s_{k'_1,k'_2}(\max(i'_1, i'_2)) D_{(k_2,i_2)(k'_2,i'_2)} &= \sum_{i'_1,i'_2} B_{i_1 i'_1} [s_{k_1 \wedge k_2}(\max(i'_1, i'_2)) - s_{f(k_1) \wedge k_2}(\max(i'_1, i'_2)) - \\
& s_{k_1 \wedge f(k_2)}(\max(i'_1, i'_2)) + s_{f(k_1) \wedge f(k_2)}(\max(i'_1, i'_2))] B_{i_2 i'_2} \qquad (51)
\end{aligned}$$

When $k_1 \neq k_2$, then (51) yields 0. Indeed, if $k_2 \succ k_1$, $k_1 \wedge k_2 = f(k_1) \wedge k_2 = k_2$, while $k_1 \wedge f(k_2) = f(k_1) \wedge f(k_2) = f(k_2)$, and hence (51) is zero. Similar for $k_1 \succ k_2$. In all other

cases, $k_1 \wedge k_2 \succ k_1, k_2$ and so $k_1 \wedge k_2 = f(k_1) \wedge k_2 = k_1 \wedge f(k_2) = f(k_1) \wedge f(k_2) = f(k_2)$ and (51) is again zero.

When $k_1 = k_2$, $k_1 \wedge k_2 = k_1$, $k_1 \wedge f(k_2) = f(k_1) \wedge k_2 = f(k_1) \wedge f(k_2) = f(k_1)$ and (51) reduces to

$$\sum_{i'_1, i'_2} B_{i_1 i'_1} [s_{k_1}(\max(i'_1, i'_2)) - s_{f(k_1)}(\max(i'_1, i'_2))] B_{i_2 i'_2} \tag{52}$$

Substituting $B_{ij} = \delta_{ij} - \delta_{ij+1}$ in (52), it is easy to verify the following:

$$\sum_{i'_1, i'_2} B_{i_1 i'_1} [s_{k_1}(\max(i'_1, i'_2)) - s_{f(k_1)}(\max(i'_1, i'_2))] B_{i_2 i'_2} = \begin{cases} \sum_{j>0} \alpha_{k_1}(j) & i_1 = i_2 = 0 \\ \alpha_{k_1}(i_1) & i_1 \neq 0,\ i_2 = 0 \\ \alpha_{k_1}(i_2) & i_2 = 0,\ i_2 \neq 0 \\ 0 & i_1 \neq i_2,\ i_1, i_2 \neq 0 \end{cases} \tag{53}$$

Multicast Inference of Packet Delay Variance at Interior Network Links

N.G. Duffield[1] F. Lo Presti[1,2]
[1]AT&T Labs–Research [2]University of Massachusetts

Abstract— **End-to-end measurement is a common tool for network performance diagnosis, primarily because it can reflect user experience and typically requires minimal support from intervening network elements. Challenges in this approach are (i) to identify the locale of performance degradation; and (ii) to perform measurements in a scalable manner for large and complex networks. In this paper we show how end-to-end delay measurements of multicast traffic can be used to estimate packet delay variance on each link of a logical multicast tree. The method does not depend on cooperation from intervening network elements; multicast probing is bandwidth efficient. We establish desirable statistical properties of the estimator, namely consistency and asymptotic normality. We evaluate the approach through model based and network simulations. The approach extends to the estimation of higher order moments of the link delay distribution.**

Keywords— **End-to-end measurement, queueing delay, estimation theory, multicast trees, network tomography**

I. Introduction

A. Background and Motivation.

Monitoring the performance of large communications networks and diagnosing the causes of its degradation is a challenging problem. There are two broad approaches to performance diagnosis. In the *internal* approach, direct measurements are made at or between network elements, e.g. of packet loss or delay. This approach has a number of potential limitations: it may not be available for general users; coverage may not span paths of interest; measurements may be disabled during period of high load; issues of scale gathering and correlating the measurements in large networks; how to compose per hop measurements to and end-to-end view.

This motivates *external* approaches, diagnosing the network through end-to-end measurements, without necessarily assuming the cooperation of network elements on the path. There has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [1], [2], [8], [21], [16], [23], [24], [26]); several measurement infrastructure projects are in development (including CAIDA [6], Felix [10], IPMA [12], NIMI [15], Surveyor [30]) with the aim to collect and analyze end-to-end measurements across a mesh of paths between a number of hosts. Standard diagnostic tools for IP networks, `ping` and `traceroute` report roundtrip loss and delay. A recent refinement of this approach, `pathchar` [13], estimates hop-by-hop link capacities, packet delay and loss rates. `pathchar` is still under evaluation; initial experience indicates many packets are required for inference leading to either high load of measurement traffic or long measurement intervals, although adaptive approaches can reduce this [9]. More broadly, measurement approaches based on Time To Live (TTL) expiry require the cooperation of network elements in returning Internet Control Message Protocol (ICMP) messages. In future, encapsulation may hide TTL from higher layers that would see just a single hop between tunnel endpoints. Finally, the success of active measurement approaches to performance diagnosis may itself cause increased congestion if intensive probing techniques are widely adopted.

In response to some of these concerns, a multicast-based approach to active measurement has been proposed recently in [3], [4]. The idea is that correlation in performance seen on *intersecting* end-to-end paths can be used to draw inferences about the performance characteristics of their common portion, without cooperation from the network. Multicast traffic is well suited for this since a given packet only occurs once per link in the (logical) multicast tree. Characteristics such as loss and end-to-end delay seen at different endpoints are highly correlated. Another advantage is in scalability. Suppose packets are exchanged on a mesh of paths between a collection of N measurement hosts stationed in a network. With unicast the probe load on the network may grow proportionally to N^2 in some links of the network. with multicast the load grows proportionally only to N.

B. Contribution

In this paper we describe a method to infer the variance of internal link delays from measured end-to-end delays

This work was sponsored in part by the DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

Address: AT&T Labs–Research, 180 Park Avenue, Florham Park, NJ 07932, USA; E-mail: {duffield,lopresti}@research.att.com

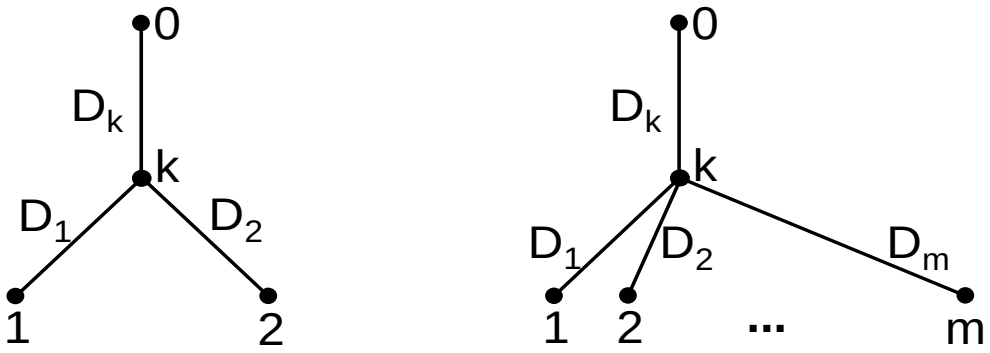


Fig. 1. LEFT: Two leaf tree. RIGHT: m-leaf tree.

of multicast probe packets. It is assumed that the link delays are independent random variables, both spatially (i.e. between different links) and temporally (i.e. between different packets); later we discuss the impact of violation of these assumptions. The method rests on (generalizations of) the following observation. Consider the logical multicast topology of Figure 1(left), in which packets are multicast from the root 0 to receivers 1 and 2. D_i is the random delay on link i, and the source-to-leaf delays from the root 0 to the leaf nodes 1 and 2 are $X_1 = D_k + D_1$ and $X_2 = D_k + D_2$ respectively. Then a simple calculation shows that, under the independence assumption,

$$\mathsf{Var}(D_k) = \mathsf{Cov}(X_1, X_2), \tag{1}$$

i.e. we express the variance of an internal link delay in terms of the covariance of the source-to-leaf delays. We can form an unbiased estimate of $\mathsf{Cov}(X_1, X_2)$ directly from end-to-end measurements; this constitutes an unbiased estimate of $\mathsf{Var}(D_k)$. The same method extends to higher order moments; when the node k had branching ratio m, we are able to estimate the first m moments of D_k; see Figure 1(right). We specify the delay model in Section II and describe the basic moment estimators in Section III.

Here we focus on estimation of the delay variance, either on individual links, or from the root to a given node. In Section IV we show how the above scheme can be used to obtain multiple unbiased estimates of the variance of the delay from the root to a given node k, one estimator for every pair of leaf nodes descended through different children of k. The estimates are consistent, i.e., they converge in probability to the true variance as the number of probes grows to infinity. Any convex combination of these estimators shares these properties; although the rate of convergence will be different in each case. This rate can be used to distinguish between the estimators. We show how to choose the weights in order to obtain the combination with the fastest asymptotic rate of convergence.

Packet loss reduces the number of packets available for delay estimation, hence increasing estimator variance. In Section V we quantify this for an estimation scheme that makes maximal use of information from surviving packets, using all packets reaching a given node pair for which a covariance estimator is calculated.

The model used here also assumes temporal independence, i.e., that delays between successive probe packets at a given node are not dependent. This can be arranged for by making the interprobe times greater than the queueing timescale. However, for a wide class of temporally dependent delay processes–we require only ergodicity–the consistency of the estimators is unaffected, i.e., they still converge to the true values as the number of probes grows to infinity. However, the rate of convergence may be slower.

In Section VI we report two types of simulation (i) model simulations with packet delay chosen pseudo-randomly according to a given distribution; and (ii) **ns** [22] simulations that represented both the probe traffic mixed in with background traffic of TCP and UDP sessions and delay occurred as result of queueing against background traffic, and loss due to buffer overflow. The model simulations allow us to compare the theoretical prediction with a model in a controlled manner. We verify the accuracy of the delay variance estimator. The variance of the variance estimators over many simulation runs is conformant with the model; this verifies the benefit in accuracy of using the minimum variance estimator. The **ns** simulation allow us to investigate the performance of the inference method in a more realistic setting in which the independence assumption may not be exactly satisfied. We find that dependence between delays in different links is smaller when buffers are larger, and that inference is correspondingly more accurate. In a 12 node topology we find the typical error in estimation is about 23%, based on a sample size of 1,000 probes. We believe this is sufficiently accurate to distin-

guish links with high delay variance. As far as we are aware there are no studies in deployed networks that measure delay correlation between different nodes. However, we believe that large and long-lasting spatial dependence is unlikely in a real network such as the Internet because of its traffic and link diversity.

C. Implementation Requirements

Since the data for delay inference comprises one-way packet delays, the primary requirement is the deployment of measurement hosts with synchronized clocks. (Actually, since delay covariances are invariant under time-shifts, the absolute times need not be synchronized, provided that the rates are identical). Using Global Positioning System (GPS) timing it is possible to make one-way delay measurements accurate to within tens of microseconds or better. GPS is currently used or planned in several of the measurement infrastructures mentioned earlier. The Network Time Protocol (NTP) [17] is more widely deployed, but provides accuracy in only the order a few tens of milliseconds, a resolution at least as coarse as the queueing delays in practice. An alternative approach to calibration and synchronization of clocks has been developed in [25], [27], [18].

Another requirement is to know the multicast topology. There is a multicast-based measurement tool, `mtrace` [19], already in use in the Internet. `mtrace` reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss and rate. Presently it does not support delay measurements. A potential drawback for larger topologies is that `mtrace` does not scale to large numbers of receivers as it needs to run once for each receiver to cover the entire multicast tree. In addition, `mtrace` relies on multicast routers responding to explicit measurement queries; the feature that can be administratively disabled. An alternative approach that is closely related to the work on multicast-based loss inference [3], [4] is to infer the logical multicast topology directly from measured probe statistics; see [5], [28]. The delay variance estimates of the present paper can also be used to infer topology. This method does not require cooperation from the network.

D. Use of Delay Variance Estimate

Although prior work has characterized end-to-end delays [1], [21], [24], to the best of our knowledge there is no generally accepted model for per link delays in real networks. Without a model it is difficult to map a given inferred value of the link delay variance to a specific value of a quality metric, such as the probability of queueing delay exceeding a given value. Nevertheless, we believe that knowledge of the per link delay variance will be increasingly useful for the following reasons:

Model Development. The mapping problem just described will become easier upon development of delay models. We expect these to arise from two sources. The first is the development of measurement infrastructure projects in which selected links are instrumented for one-way delay measurements. The second is the development of multicast-based estimators for the link delay *distribution* from end-to-end measurements, using a more computationally intensive technique proposed in a companion paper [14]. We anticipate that this will allow the development of link delay distribution models, with the distribution inferred from network measurements.

Ordering. Identification of links with highest delay variance suggests candidate for links on which performance is degraded for delay sensitive applications.

Delay and Delay Variation. The variance of the packet delay (on a link or path) can be used to estimate or bound the variance of the interpacket delay variation. Let D^i be the delay encountered by packet i on a given link. The interpacket delay variation (or jitter) between packets i and $i+1$ on the link is $J^i = D^{i+1} - D^i$; a similar notion applies to end to end delay. Observe

$$\mathsf{Var}(J^i) = \mathsf{Var}(D^i) + \mathsf{Var}(D^{i+1}) - 2\mathsf{Cov}(D^i, D^{i+1}). \quad (2)$$

Assuming $D(\cdot)$ to be stationary, the first two terms on the RHS of (2) are equal, while under the assumption of temporal independence the last term is zero, and so $\mathsf{Var}(J^i) = 2\mathsf{Var}(D^i)$. Measurements of end-to-end delays in the Internet [1] show that end-to-end delays successive packets are only slightly dependent when the interpacket time is longer than the typical queueing timescales. Stronger dependence is found at shorter timescales: successive packets are more likely to queue together. With positive correlation between successive probe delays $\mathsf{Cov}(D^i, D^{i+1}) > 0$; in this case $\mathsf{Var}(J^i)$ is bounded above by $2\,\mathsf{Var}(D^i)$, a quantity that we can estimate from end-to-end measurements.

Topology Inference. If the logical multicast topology is not initially known, it can be inferred from delay variances. This technique uses the estimated variance of the cumulative delay from the source to a given node. Consequently we shall be interested here in the estimation of cumulative delay variance as well as link delay variance.

II. The Tree and Delay Models

We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links than join them. The logical multicast tree comprises

the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except the leaf nodes and possibly the root, must have 2 or more children. We can construct the logical tree from the physical tree by deleting all links with one child and adjusting the links accordingly by directly joining its parent and child.

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify one node, the root 0, with the source of probes, and $R \subset V$ will denote the set of leaf nodes (identified as the set of receivers). The set of children of node $j \in V$ is denoted by $d(j)$. Each node, k, apart from the root has a parent $f(k)$ such that $(j,k) \in L$. Define recursively the compositions $f^n = f \circ f^{n-1}$ with $f^1 = f$. Nodes are said to be siblings if they have the same parent. If $k = f^m(j)$ for some $m \in \mathbb{N}$ we say that j is descended for k (or equivalently that k is an ancestor of j) and write the corresponding partial order in V as $j \prec k$. $i \vee j$ will denote the minimal common ancestor of i and j in the $\preceq$-ordering.

We associate each node k a random variable D_k taking values in the extended positive real line $\overline{\mathbb{R}} = \mathbb{R}_+ \cup \{\infty\}$. By convention $D_0 = 0$. D_k is the random delay that would be encountered by a packet attempting to traverse the link $(f(k), k) \in L$. The value $D_k = \infty$ indicates the packet is lost on the link. The delay experienced on the path from the root 0 to a node k is $X_k = \sum_{j \preceq k} D_j$. We assume that the D_k are independent. Let $\alpha_k = \mathsf{P}[D_k < \infty]$, the probability of successful transmission over link k.

III. Non-parametric Estimation of Delay Distribution Moments

In this section we present a class of non-parametric estimators of the delay distribution. We assume initially that all delays are finite: $\mathsf{P}[D_k = \infty] = 0$. Consider first a logical subtree formed by the root 0, and a non-leaf node k with two descendents 1 and 2 that are leaf nodes; see Figure 1(left). By writing $X_i = X_k + (X_i - X_k)$ in the expression for $\mathsf{Cov}(X_1, X_2)$, expanding using the bilinearity of the covariance operator $\mathsf{Cov}(\cdot,\cdot)$, and using the mutual independence of the links delays $X_k, X_1 - X_k$ and $X_2 - X_k$, we obtain

$$\mathsf{Cov}(X_1, X_2) = \mathsf{Var}(X_k). \tag{3}$$

Hence any unbiased estimator of $\mathsf{Cov}(X_1, X_2)$ is also an unbiased estimator of $\mathsf{Var}(X_k)$. Let $X_1^{(i)}, X_2^{(i)}$, $i = 1, 2, \ldots n$ be measured end-to-end delays between the root 0 and leaf nodes 1 and 2 respectively. Abbreviate $\mathsf{Cov}(X_j, X_k)$ by s_{jk} and write s_{kk} as s_k. We estimate s_k by a uniformly minimum variance unbiased estimator of s_{12}, namely $\widehat{s}_{12}$ where

$$\widehat{s}_{ij} = \frac{1}{n-1}\left(\sum_{m=1}^{n} X_i^{(m)} X_j^{(m)} - \frac{1}{n}\sum_{m,m'=1}^{n} X_i^{(m)} X_j^{(m')}\right) \tag{4}$$

At a node with branching ratio m we are able to estimate the first m moments of the delay on the shared portion of the path from the root; see Figure 1(right). The cumulant generating function of the m leaf delays $\boldsymbol{X} = (X_1, \ldots X_m)$ is defined for $\boldsymbol{\theta} \in \mathbb{R}^m$ by

$$\lambda(\boldsymbol{\theta}; \boldsymbol{X}) = \log \mathsf{E}[\exp(\sum_{i=1}^{m} \theta_i X_i)]. \tag{5}$$

The cumulants are defined by partial differentiation w.r.t. the components θ_i (when derivatives exist): for indices $j_1, \ldots, j_m \in \mathbb{Z}_+$ set

$$K^{j_1,\ldots,j_m}(\boldsymbol{X}) = \left(\prod_{i=1}^{m} \frac{\partial^{j_i}}{\partial \theta_i^{j_i}}\right) \lambda(\boldsymbol{\theta}; \boldsymbol{X})\Big|_{\theta_i = \mathbf{0}} \tag{6}$$

The first and second cumulants K^1 and K^2 of a single random variable are its mean and variance respectively. Knowing the cumulants of a set of random variables is equivalent to knowing their joint distribution. The cumulants of D_0 are related to those of the X_i as follows. Set $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^m$.

Theorem 1: $K^{\mathbf{1}}(\mathbf{X}) = K^m(X_k)$. Hence any unbiased estimator of $K^{\mathbf{1}}(\mathbf{X})$ is also an unbiased estimator of $K^m(X_k)$.

Proof: Observe $K^{\mathbf{1}}(X_1, \ldots, X_m) = K^{\mathbf{1}}(X_1 - X_k, \ldots, X_m - X_k) + K^{\mathbf{1}}(\mathbf{1}X_k) = K^m(X_k)$. The first equality is because K is affine in each of its arguments, the second because the cumulant of a set of independent random variables is zero. ■

IV. Delay Variance Estimation on General Trees

In a general tree let $Q(k) = \{\{i,j\} \subset R \mid i \vee j = k, \}$ be the set of distinct pairs of leaf-nodes whose $\prec$-least common ancestor is k. Any convex combination $\sum_{\{i,j\} \in Q(k)} \mu_{ij} \widehat{s}_{ij}$ (i.e. with the $\mu_{ij} \geq 0$ and summing to 1) is also an unbiased estimator of s_k. An example the **uniform estimator**

$$\frac{1}{\#Q(k)} \sum_{\{i,j\} \in Q(i \vee j)} \widehat{s}_{ij}. \tag{7}$$

One potential disadvantage with the uniform estimator is that high variance of one of the summands may lead to

high estimator variance overall. This motivates choosing convex combinations that are functions of the end-to-end delays themselves in order to reduce variance. In this section we shall assume that all delays a finite with bounded fourth moments. We shall relax the finiteness assumption in Section V.

We formalize the notion of (possibly random) convex combinations of $\widehat{s}_{ij}$ through a **covariance aggregator**. For $S \subset R$ let $\mathcal{F}_n(S)$ denote the σ-algebra generated by the end-to-end delays $(X_k)_{k\in S}$ (i.e. the set of events that can be determined from knowing $(X_k)_{k\in S}$). A covariance aggregator μ is sequence $(\mu(n))_{n\in\mathbb{N}}$ of random vectors $\{\mu_{ij}(n) : \{i,j\} \in Q(k);\ k \in V\setminus R\}$ with $0 \le \mu_{ij}(n) \le 1$ and $\sum_{\{i,j\}\in Q(k)} \mu_{ij}(n) = 1$ for each $k \in V \setminus R$. We assume each μ_n to be $\mathcal{F}_n(R)$-measurable, i.e., it is a function of the measured delays of the first n probes. We will usually suppress the explicit dependence on the number of probes n. Let $\widehat{s} = \{\widehat{s}_{ij}(n) : \{i,j\} \in Q(k);\ k \in V\setminus R\}$ be a family of estimators, $\widehat{s}_{ij}(n)$ being an $\mathcal{F}_n(i,j)$-measurable unbiased estimator of $\mathsf{Var}(X_{i\vee j})$. Then we estimate $\mathsf{Var}(X_k)$ by

$$V_k(\mu,\widehat{s}) = \sum_{\{i,j\}\in Q(k)} \mu_{ij}\widehat{s}_{ij} \tag{8}$$

A covariance aggregator is called **deterministic** if it does not depend on the $X^{(i)}$. We denote the set of such aggregators with indices in $Q(k)$ by $\mathcal{D}_k$. An example is the **uniform** aggregator that was used in the uniform estimator (7): $\mu_{ij} = (\#Q(i\vee j))^{-1}$. Define the covariance matrix

$$C_{(ij),(\ell m)} = \mathsf{Cov}\left(\ Z_iZ_j\ ,\ Z_\ell Z_m\ \right), \tag{9}$$

where $Z_i = X_i - \mathsf{E}[X_i]$. We will use $C(k) = \left[C_{(ij),(\ell m)}\right]_{(ij),(\ell m)\in Q(k)}$ to denote the matrix obtained by letting the indices (ij) and (ℓm) in (9) run over $Q(k)$; this is a submatrix of the matrix $C^0(k)$ obtained by taking the indices unrestricted over the set $Q^0(k)$ of binary subsets of $R(k)$.

A. Minimum Variance Estimation for Cumulative Delays

In the next theorem we characterize the asymptotic distribution of the $\widehat{s}_{ij}$ as $n\to\infty$, and give a form for the estimator $V_k(\mu,\widehat{s})$ of minimum cumulative variance.

Theorem 2: (i) For each $k \in V\setminus R$ the random variables $\{\sqrt{n}\ (\widehat{s}_{ij} - s_k) \mid \{i,j\} \in Q(k)\}$ converge in distribution as $n\to\infty$ to a multivariate Gaussian random variable with mean 0 and covariance matrix $C(k)$. Hence the $\widehat{s}_{ij}$ are consistent estimators of s_k and so is $V(\mu,\widehat{s})$. For any deterministic covariance aggregator μ, $\sqrt{n}(V_k(\mu,\widehat{s}) - s_k)$ converges in distribution as $n\to\infty$ to a Gaussian random variable of mean zero and variance $\mu\cdot C(k)\cdot\mu$.

(ii) The minimal asymptotic variance $\inf_{\mu\in\mathcal{D}_k}\mu\cdot C(k)\cdot\mu$ is achieved when

$$\mu_{ij} = \mu^*_{ij}(C(k)) := \left(C(k)^{-1}\cdot\mathbf{1}\right)_{(ij)} \Big/ \mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1} \tag{10}$$

where $C(k)^{-1}$ denotes the inverse matrix of $C(k)$ and $\mathbf{1}_{(ij)} = 1$, $\{i,j\}\in Q(k)$. The corresponding asymptotic variance of the variance estimator is $\left(\mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1}\right)^{-1}$.

Proof: (i) The proof follows from standard results in multivariate analysis; convergence to the stated Gaussian random variable follows by Corollary 1.2.18 in [20]

(ii) Since the μ_{ij} sum to 1, the proof follows by considering the constrained minimization of $\mu\cdot C(k)\cdot\mu - 2k\mu\cdot\mathbf{1}$ with Lagrange multiplier k. As a covariance matrix, $C(k)$ is positive definite and hence invertible; minimization of the convex function of μ takes place at the the stationary point $\mu = kC(k)^{-1}\cdot\mathbf{1}$. This yields $\mu^*(C(k))$ upon normalization. The corresponding minimal asymptotic variance is $\mu^*(C(k))\cdot C(k)\cdot\mu^*(C(k)) = \left(\mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1}\right)^{-1}$. ■

Operationally, the coefficients μ_{ij} of the minimum variance estimator $V_k(\mu^*(C(k)),\widehat{s})$ of Theorem 2 are to be calculated from an *estimate* of the covariance matrix $C(k)$. Let $Z_i^{(m)} = X_i^{(m)} - \frac{1}{n}\sum_{m=1}^n X_i^{(m)}$. Let $\widehat{C}(k)$ denote the empirical covariance matrix with entries

$$\widehat{C}(k)_{(ij),(i'j')} = \frac{n^2}{(n-1)^3}\left(\sum_{m=1}^n Z_i^{(m)}Z_j^{(m)}Z_{i'}^{(m)}Z_{j'}^{(m)} - \frac{1}{n}\sum_{m=1}^n Z_i^{(m)}Z_j^{(m)}\sum_{m=1}^n Z_{i'}^{(m)}Z_{j'}^{(m)}\right) \tag{11}$$

$\widehat{C}(k)$ is an unbiased estimator of $C(k)$. Estimating $\mu^*(C(k))$ by $\mu^*(\widehat{C}(k))$ and s_k by $V_k(\mu^*(\widehat{C}),\widehat{s})$ potentially introduces bias and increases variance in the estimation of the s_k. However, the following Theorem shows that it is consistent and has the same asymptotic variance as $V_k(\mu^*(C),\widehat{s})$.

Theorem 3: $V_k(\mu^*(\widehat{C}(k)),\widehat{s})$ is a consistent estimator of s_k. $\sqrt{n}(V_k(\mu^*(\widehat{C}(k),\widehat{s}) - s_k)$ converges in distribution to a Gaussian random variable of mean zero and variance $\left(\mathbf{1}\cdot C(k)^{-1}\cdot\mathbf{1}\right)^{-1}$.

Proof: Clearly $\widehat{C}(k)$ converges almost surely to $C(k)$ as $n\to\infty$. Since matrix inversion is continuous on the set of strictly positive definite matrices, $\mu^*(\widehat{C}(k))$ converges almost surely (to $\mu^*(C(k))$); since each $\widehat{s}_{ij}$ converges to $s_{ij} = s_k$, $V_k(\mu^*(\widehat{C}(k)),\widehat{s})$ is consistent.

By the δ-method (see e.g. [29]), $\sqrt{n}\ (V(\mu^*(\widehat{C}(k)),\widehat{s}) - s_k)$ converges to a Gaussian random variable with mean 0

and variance $\alpha \cdot C^0(k) \cdot \alpha$, where for $(\ell, m) \in Q^0(k)$,

$$\alpha_{\ell m} = \frac{\partial}{\partial s_{\ell m}} \sum_{\{i,j\} \in Q(k)} \mu^*_{ij}(C(k)) s_{ij}. \qquad (12)$$

Differentiating,

$$\begin{aligned} \alpha_{\ell m} &= \mu^*_{\ell m}(C(k)) \chi_{Q(k)}(\{\ell, m\}) \\ &+ \sum_{\{i,j\} \in Q(k)} s_{ij} \frac{\partial}{\partial s_{\ell m}} \mu^*_{ij}(C(k)), \end{aligned} \qquad (13)$$

where $\chi_{Q(k)}$ denotes the indicator function of the set $Q(k)$. But $s_{ij} = s_{i \vee j} = s_k$ for $\{i, j\}$ in $Q(k)$ and so is constant in the sum. Since the μ^*_{ij} sum to 1, the sum in (13) is zero. Hence $\alpha \cdot C^0(k) \cdot \alpha = \mu^*(C(k)) \cdot C(k) \cdot \mu^*(C(k))$. ∎

B. Minimum Variance Estimation for Link Delays

We can estimate the link delay variance as the difference of two cumulative variances since

$$\mathsf{Var}(X_k) = \mathsf{Var}(X_{f(k)} + D_k) = \mathsf{Var}(X_{f(k)}) + \mathsf{Var}(D_k), \qquad (14)$$

by the independence assumption on link delays. An unbiased estimator of $r_k := \mathsf{Var}(D_k)$ is $V_k(\mu^*(C(k)), \widehat{s}) - V_{f(k)}(\mu^*(C(f(k))), \widehat{s})$. We now show that joint optimization of the aggregators in V_k and $V_{f(k)}$ will result in an estimator of lower variance.

Given a pair $\mu = (\mu(k), \mu(f(k))) \in \mathcal{D}_k \times \mathcal{D}_{f(k)}$ of deterministic covariance aggregators with indices in $Q(k)$ and $Q(f(k))$ respectively, we can form a unbiased estimate of r_k as

$$W_k(\mu, \widehat{s}) := V_k(\mu(k), \widehat{s}) - V_{f(k)}(\mu(f(k)), \widehat{s}) \qquad (15)$$

Let $C'(k)$ denote the $\#Q(k) + \#Q(f(k))$ dimensional matrix written in block form

$$C'(k) = \begin{pmatrix} C(k) & C(k, f(k)) \\ C(k, f(k))^T & C(f(k)) \end{pmatrix}, \qquad (16)$$

where $C(k, f(k))$ is the $\#Q(k) \times \#Q(f(k))$ matrix of covariances $\left[C_{(ij),(\ell m)}\right]_{(ij) \in Q(k), (\ell m) \in Q(f(k))}$. Then statements analogous to Theorem 2(ii) follow straightforwardly, using parallel arguments. In particular $\sqrt{n}(W_k(\mu, \widehat{s}) - r_k)$ converges to a Gaussian random variable of mean 0 and variance $\mu \cdot C'(k)^{-1} \mu$ and the minimum over deterministic aggregators of the asymptotic variance takes the value $(c_1 + c_2 + 2c_3)/(c_1 c_2 - c_3^2)$ where $c_1 = \mathbf{1}_k \cdot C(k)^{-1} \cdot \mathbf{1}_k$, $c_2 = \mathbf{1}_{f(k)} \cdot C(f(k))^{-1} \cdot \mathbf{1}_{f(k)}$ and $c_3 = \mathbf{1}_{f(k)} \cdot C(k, f(k))^{-1} \cdot \mathbf{1}_k$. (Here the subscripts on $\mathbf{1}_k, \mathbf{1}_{f(k)}$ distinguish the subspaces in which these vectors live).

C. Criteria for Assessing Inference Reliability

In sections IV-A and IV-B we derived expressions for the variances of estimates of the cumulative and link delays respectively. For a given delay variance estimate, we can associate its variance by using the plug in estimator for the corresponding analytic expression. This enables use to find confidence intervals for the estimates that will be asymptotically accurate for large n. For example, if we use n probes to form the estimate $V_k(\mu^*(\widehat{C}(k)), \widehat{s})$, we associate with this a variance σ^2/n where $\sigma^2 = (\mathbf{1} \cdot \widehat{C}(k)^{-1} \cdot \mathbf{1})^{-1}$. We write confidence limits for the estimate as

$$V_k(\mu^*(\widehat{C}(k), \widehat{s}) \pm z_{\delta/2}\, \sigma/n, \qquad (17)$$

where $z_{\delta/2}$ denotes the number that cuts off an area $\delta/2$ in the right tail of the standard normal distribution. This is used for a confidence interval of level $1 - \delta$.

V. Impact of Loss on Estimator Variance

We relax the assumption of finite delays, Here we identify infinite delays with packet loss, although the same results would hold were we to treat as lost any packet with source to leaf delay greater than some finite value. The link and cumulative delay random variables will be denoted by D'_k and X'_k respectively each possibly taking the value ∞. We use D_k to be the distribution of D'_k conditional on $D'_k < \infty$, and similarly for X_k. We assume throughout that the D_k have finite fourth moments. Since we are interested in delay variance, we want to estimate $\mathsf{Var}(X_k)$ and $\mathsf{Var}(D_k)$ even in the presence of packet loss. For estimation, the effect of packet loss is to reduce the number of delay samples available, and hence to increase the variability of the estimates. A simple way to apply the foregoing theory is to restrict attention to only those packets that are received at every leaf (or at least at every element of $R(k)$ when estimating s_k). A disadvantage of this approach is that is does not scale well as the topology grows. For assuming link loss rates to be bounded away from zero, the proportion of packets reaching all receivers in a tree decays geometrically fast in the number of links in the tree.

An alternative that wastes less data is to calculate pairwise estimates of $\widehat{s}_{ij}$ that use all packets received at i and j. Let us formalize this. For a subset of receivers $S \subset V$ define $I_n(S) = \{i \in \{1, 2, \ldots, n\} \mid X_j^{(i)} < \infty \ \forall j \in S\}$: the subset of the first n probes that are received at all nodes in S; set $N_n(S) = \#I_n(S)$. We will sometimes write $I_n(i_1, \ldots, i_r)$ for $I_n(\{i_1, \ldots, i_r\})$, and similarly for N_n. For $S \subset R$ let $V(S)$ be the set of nodes in the minimal tree spanning 0 and S. Set $B(S) = \prod_{i \in V(S)} \alpha_i$, where α_i is the probability of successful transmission over link

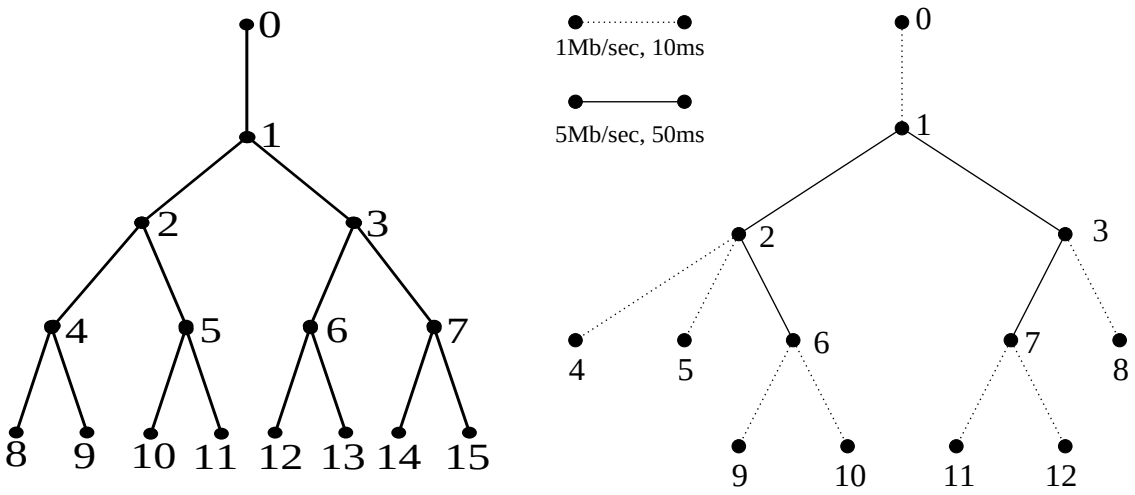


Fig. 2. TREES USED IN SIMULATIONS. LEFT: 8-leaf binary tree for model simulations ; RIGHT: Heterogeneous 7-leaf tree for **ns** simulation.

k. Clearly $n^{-1}N_n(S)$ converges almost surely to $B(S)$ as $n \to \infty$. Estimator variance can be reduced by using all packets in $I_n(i,j)$ to estimate s_{ij}, not just those in $I_n(R(i \vee j)$. Define

$$\widehat{v}_{ij} = \frac{1}{N}\left(\sum_m X_i^{(m)} X_j^{(m)} - \frac{1}{N}\sum_{m,m'} X_i^{(m)} X_j^{(m')}\right) \tag{18}$$

where N abbreviates $N_n(i,j)$ and in the sums m, m' run over $I_n(i,j)$. $\widehat{v}_{ij}$ is an unbiased estimate on s_{ij}. Analogous to the previous results we have

Theorem 4: (i) For each $k \in V \setminus R$ the random variables $\{\sqrt{n}\ (\widehat{v}_{ij} - s_k) \mid \{i,j\} \in Q(k)\}$ converge in distribution as $n \to \infty$ to a multivariate Gaussian random variable with mean 0 and covariance matrix $G(k)_{(ij),(\ell m)} = C(k)_{(ij),(\ell m)} B(i,j,\ell,m)/(B(i,j)B(\ell,m))$. Hence the $\widehat{v}_{ij}$ are consistent estimators of s_k and so is $V_k(\mu, \widehat{v})$ for any deterministic covariance aggregator μ. For any deterministic covariance aggregator μ, $\sqrt{n}(V_k(\mu,\widehat{v}) - s_k)$ converges in distribution as $n \to \infty$ to a Gaussian random variable of mean zero and variance $\mu \cdot G(k) \cdot \mu$.

(ii) The minimal asymptotic variance $\inf_{\mu \in \mathcal{D}_k} \mu \cdot G(k) \cdot \mu$ is achieved when $\mu = \mu^*(G)$; the corresponding minimal asymptotic variance is $\left(\mathbf{1} \cdot G(k)^{-1} \cdot \mathbf{1}\right)^{-1}$.

(iii) $V_k(\mu^*(\widehat{G}), \widehat{v})$ has the same asymptotic properties as $V_k(\mu^*(G), \widehat{v})$ where the estimated covariance $\widehat{G}$ is defined by

$$\begin{aligned}\frac{N_n(i,j)N_n(k,\ell)}{N_n(i,j,k,\ell)}\widehat{G}_{(ij),(k\ell)} &= \sum_m Z_i^{(m)} Z_j^{(m)} Z_k^{(m)} Z_\ell^{(m)} \\ &\quad - \frac{1}{N_n(i,j,k,\ell)} \sum_m Z_i^{(m)} Z_j^{(m)} \sum_{m'} Z_k^{(m)} Z_\ell^{(m')}\end{aligned} \tag{19}$$

where the sums run over $I_n(i,j,k,\ell)$.

The corresponding version of the minimum variance link delay variance estimator follows by replacing C by G and $\widehat{s}$ by $\widehat{v}$ throughout Section IV-B.

VI. SIMULATION EVALUATION

We conducted two types of simulation (i) model simulation with packet delay chosen pseudo-randomly according to a given distribution; and (ii) **ns** [22] simulations that represented both the probe traffic mixed in with background traffic of TCP and UDP sessions and delay occurred as result of queueing against background traffic, and loss due to buffer overflow. The model simulations allow us to compare the theoretical prediction with a model in a controlled manner; their purpose is to show that the statistical properties of the estimators conform to the model used. The **ns** simulation allow us to investigate the performance of the inference method in a more realistic setting in which the model assumption (such as independence) may not be exactly satisfied. Their purpose is to investigate conformance of the predicted delay variances with those occurring in the network interior.

A. Model Simulations

The model simulation used an 8 leaf binary tree (see Figure 2(left)); delays were exponentially distributed. The delay variances were heterogeneous: leaf links 8 and 15 had delay variance 10, all other links had delay variance 1. Losses were not modeled. This heterogeneity was chosen in order to evaluate the advantages of the minimum variance estimator. We present a representative set of results from experiments for the link delay variance W and the cumulative delay variance V.

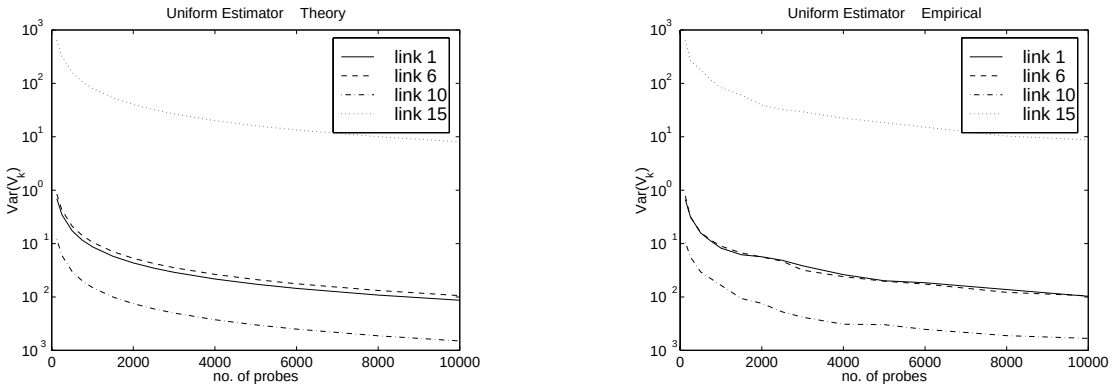


Fig. 3. VARIANCE OF ESTIMATED VARIANCE. Cumulative Delay Variance to nodes 1,6,10,15 in Figure 2(left). LEFT: Calculate Variance; RIGHT: Empirical Variance from 100 simulations.

Weight μ_{ij}	Link pairs (i,j)
0.000018	(8,15)
0.001213	(8,12) (8,13) (10,15) (11,15)
0.001811	(8,14) (9,15)
0.081286	(10,12) (10,13) (11,12) (11,13)
0.121322	(9,12) (9,13) (10,14) (11,14)
0.181077	(9,14)

TABLE I

WEIGHTS FOR MINIMUM VARIANCE ESTIMATOR. Topology of Figure 2. Links 8 and 15 have ten times variance of others.

A.1 Convergence

Figure 3 shows the variance of the cumulative delay variance from sources to nodes $k = 1, 6, 10, 15$ in Figure 2(left), plotted as a function of the number of probes. On the left is the theoretical variance $\mathsf{Var}(V_k(\mu^*(C), \widehat{s}))$; on the right the empirical variance from 100 samples of $V_k(\mu^*(C), \widehat{s})$ found by simulation. Observe in both cases the decay of the variance towards 0 as the number of probes increases; furthermore the experimental variance is very close to the theoretical values over the range of probe numbers.

Figure 4 shows detail from a single simulation; sample paths of the link variance estimator $W_k(\mu, \widehat{s})$ for links $k = 1, 3, 5, 10$ as function of the number of probes, for up to 10,000 probes. On the left figure, the aggregator μ is uniform, on the right, the minimum variance aggregator $\mu^*(\widehat{C})$. Observe in both cases that the estimate approaches the model value, 1, as the number of probes increases.

A.2 Variance Reduction

In Figure 4, convergence is tighter for the minimum variance estimator (on the right) than in the uniform case; this is particularly apparent in the left region of each plot, corresponding to smaller numbers of probes. The difference is particularly evident for link 1 (which has 2 high variance links as descendents, 8 and 15) and link 3, which has link 15 as a descendent. The variance of the estimators W_k for both these links is decreased in the minimum variance estimator, relative to the uniform estimator, by reducing the weight μ_{ij} when i or j corresponds to a high variance link. This is particularly striking in the minimum variance estimator for link 1; we tabulate the weights $\mu_{ij}(C(1))$ in Table I. The weight for the pair $(8, 15)$ of high variance links is 10^{-4} times the highest weight, that for pair $(9, 14)$.

To see the statistics of estimator variation reduction, we display in Figure 5 the ratio of the standard deviation of the uniform estimator to the standard deviation of the minimum variance estimator, and a function of the number of probes. This in shown on the left for the cumulative variance, and on the right for the link delay variance. For the cumulative variance we display only for links, 1,2 and 3; the other internal links the uniform and minimum variance estimators are identical because there is only one term in the sum for V. The figures show that the reduction in variance is roughly uniform across a range of experiment length up to 10,000 probes. The standard deviation was roughly halved for the cumulative delay variance, and between 0.3 and 0.5 for the link delay variance. Reduction was somewhat greater for the standard deviation of the link delay variance, except for nodes 4 and 7. These nodes have only two descendants, one of which terminates a high variance link; there is no flexibility to avoid the high variance of the first term of $W_k = V_k - V_{f(k)}$.

B. Network Simulation

B.1 Methodology

The **ns** simulations used the topology in Figure 2(right). We arranged for some heterogeneity between the edges

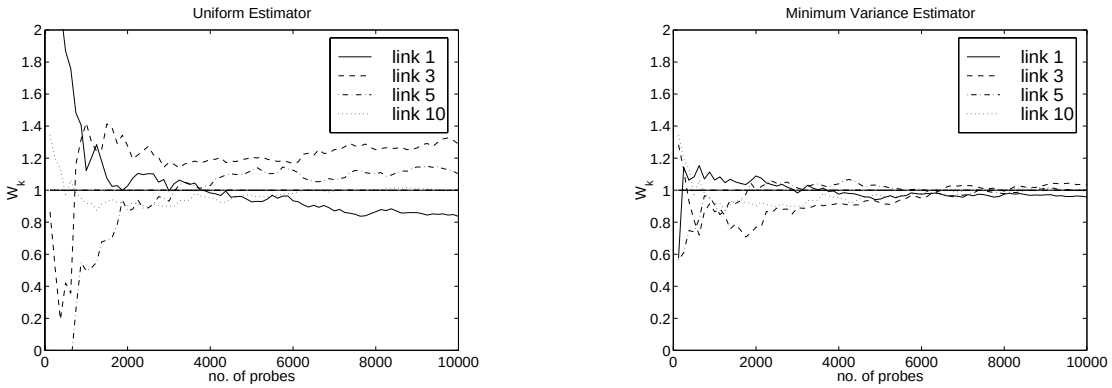


Fig. 4. SAMPLE PATHS OF LINK VARIANCE ESTIMATORS. For up to 10,000 probes. LEFT: Uniform Estimator, RIGHT: Minimum Variance Estimator

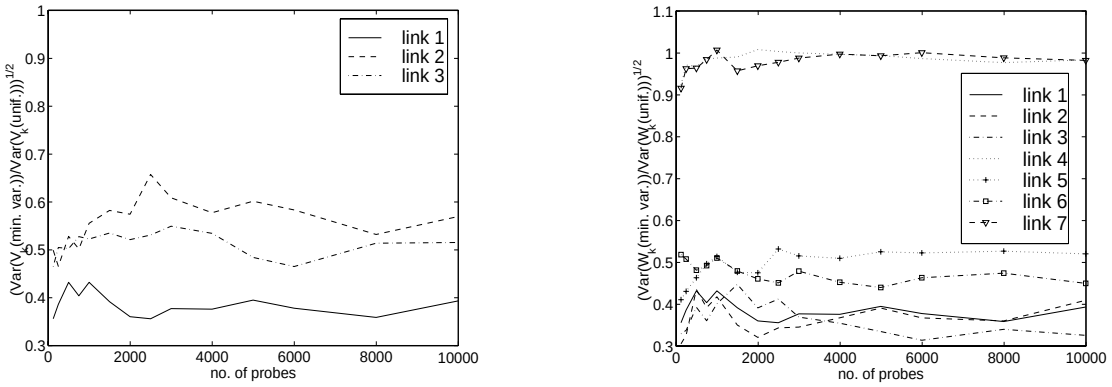


Fig. 5. VARIANCE REDUCTION Ratio of standard deviation the uniform estimator to standard deviation minimum variance estimator. LEFT: cumulative delay variance, RIGHT: link delay variance.

and the center of the tree in order to mimic the difference between the core and edges of a large WAN, with the interior of the tree having higher capacity (5Mb/sec) and latency (50ms) than at the edge (1Mb/sec and 10ms). Each node had a finite buffer capacity; packet losses were due to drops for the tail of the buffer. We used buffer capacities of 4 and 20 packets in two different sets of experiments. The cross traffic comprised 66 FTP sessions over TCP, and 29 UDP traffic sources following an exponential on-off model; there were on average around 8 background traffic sources per link. In each simulation we use the source-to-leaf delays of probes as data to infer delay variance per internal link by and also from the source to a given internal node. Since the simulations exhibit packet loss, the inference was performed using the algorithms described in Section V. We compared the inferred values W_k with the actual delay variance for probes on internal links that was observed during the simulation run. The comparison was performed over each link in Figure 2(right) for 100 simulation runs.

B.2 Comparison of Inferred and Actual Delay Variance

Figure 6 shows scatter plots of 1200 pairs of (inferred, actual) link delay variance, based on 1000 probes, on the left with buffer capacity of 4 packets, on the right with buffer capacity 20 packets. Also shown is the line through the origin at gradient 1; a point on this line would indicate an instance of perfect inference. In the scatter plots we differentiate between predictions using the uniform estimator, and those using the minimum variance estimator.

Taking each plot separately we observe that inference is more accurate for the minimum variance estimator than the uniform estimator, the difference being more evident for the smaller buffer size. Comparing the plots we see that inference is more accurate when for the simulated network with larger buffer capacities, particularly for small delay variances. A small number of inferred values were negative. This occurred for some links of high bandwidth for which queueing delays were small. Estimation of the link delay variance as the difference between the variance of the cumulative delays (see (15)) is sensitive to estimation errors. Nevertheless, the estimation error is sufficiently

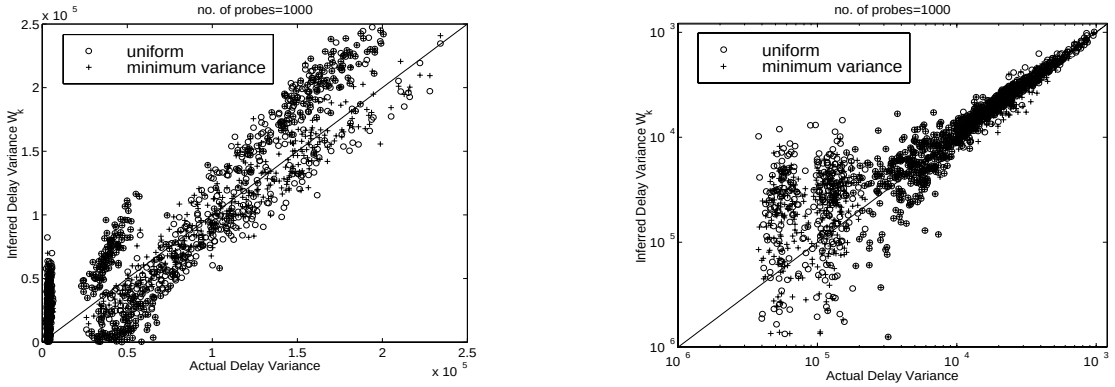


Fig. 6. SCATTER PLOTS FOR LINK DELAY INFERENCE. Pairs of (actual,inferred) over 12 links in 100 experiments, each using 1000 probes LEFT: nodes have 4 packet buffers; RIGHT: nodes have 20 packet buffers.

small that is would not impair identification of those links with the largest delay variance. Furthermore, in practice we can avoid the worst small variance estimation errors by eliminating estimates that are not significantly different from zero according at some confidence level. Similar to (17), these are the estimates W_k from n probes for which $W_k < z_\delta \sigma/\sqrt{n}$, $1-\delta$ is the desired (one-sided) confidence level, and σ^2 is the appropriate asymptotic variance expressed in terms of the estimated covariance.

We attribute bias of inference to departures of the delay process from the independence assumption of the model. We calculated the off-diagonal elements of the correlation matrix of the actual link delays. For buffer size 4 the mean value was 0.071, the maximum 0.51. For buffer size 20 the mean was 0.021, the maximum 0.17. Thus correlations were more pronounced for the smaller buffer size, leading to greater inference inaccuracy. We found that bias was more pronounced in the inference of cumulative delay, particularly for buffer size 20 where the cumulative delay variance is almost always overestimated. Bias was less evident for the link delay variance. Since this is expressed as a difference of estimated cumulative delay variance, consistent bias in the latter quantities should cancel somewhat in subtraction. Conversely, small delay variances are better estimated for for the cumulative than the link case.

In order to quantify the accuracy of inference we define a metric for evaluating estimator accuracy. If w and $\widehat{w}$ are the actual and inferred delay variances (either cumulative to a link or at the link itself) we form their *error factor*

$$F(\widehat{w}, w) = \max\left\{\frac{\widehat{w}}{w}, \frac{w}{\widehat{w}}\right\}. \qquad (20)$$

For example, if $\widehat{w}$ is either twice or half w, their error factor is 2. As a robust summary statistic to capture the center of the distribution of error factors, we use the two-sided quartile-weighted median (QWM)

$$(Q_{.25} + 2Q_{.5} + Q_{.75})/4, \qquad (21)$$

where Q_p denotes the p^{th} quantile of a given set of error factors.

	buffer = 4 pkts.		buffer = 20 pkts.	
z	unif.	min. var.	unif.	min. var.
0	2.31	2.06	1.32	1.32
2	1.56	1.76	1.23	1.23

TABLE II
QUARTILE WEIGHTED MEDIAN ERROR FACTORS FOR INFERENCE ON 1000 PACKETS. Link delay variance estimation, according to number of standard deviations z in confidence level to avoid small variances. Errors are smaller for minimum variance estimator than uniform estimator, and also with increased buffer capacity.

In Table II we display the QWM of error factors for link variance estimation. Small or negative inferred variances were omitted, the quantity z being the number of standard deviations characterizing the confidence interval about 0. $z = 0$ corresponds to rejecting only negative inferred variances. Ruling out these small variances decreases the QWM of the error factor: the smaller variances typically have higher error factor. (For $z = 2$, buffer = 4, it happens that the 75^{th}-percentile of the error factor distribution is larger for the minimum variance estimator, but this is atypical). For large buffer sizes the error factors are noticeably smaller; the difference in accuracy between the uniform and minimum variance estimator is smaller too. We found no great advantage in increasing the number of probes to 10,000 since bias becomes a larger part of the errors.

VII. Conclusions and Further Work

In this paper we have proposed a novel technique for the inference from end-to-end measurements of the variance of the delay encountered by multicast packets on an internal link. The cooperation of intervening network nodes is not required.

We constructed a convex family of variance estimators and found the estimator of minimal asymptotic variance. Evaluating the minimal variance estimator comes at some computational cost, namely, the inversion of the covariance matrix $\widehat{C}$. In work to be reported elsewhere, we show how this computation may be considerably simplified for binary trees, although at the cost of increasing estimator variance somewhat. Another approach is to compromise between the computational simplicity of the uniform estimator and variance reduction. An example would be to set $\mu_{ij} = 0$ for $\{i, j\}$ in some subset of $Q(k)$ in which the measures end-to-end variances $\widehat{s}_i$ are high. It remains to develop a robust approach along these lines.

The `ns` experiments showed typical errors of about 20% in estimation of the delay variance using 1,000 probes. We observe that using a 40 bytes probe every 100ms, the load on the network is less that 4kb/sec and the measurements can be completed within 2 minutes.

We found inference to be more accurate in networks with larger buffers; there was smaller correlation between delays at different nodes and hence closer conformance to the underlying model. It appear that the larger buffers admit a greater diversity of connections through a node over queueing timescales, diluting the correlation seen between delays at successive nodes. We believe that diversity of traffic in real networks such as the Internet makes large and long lasting correlations unlikely. Furthermore the introduction of Random Early Detection (RED) [11] policies in Internet routers may help reduce dependence; evidence for this comes from related work on internal link loss inference [4], where the introduction of RED was found to increase accuracy of inference relative to networks with a Drop from Tail packet discard mechanism.

Acknowledgment

We thank Joseph Horowitz for his suggestion to find the minimum variance estimator.

References

[1] J. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet." *Journal of High-Speed Network*, vol. 2 n. 3, pp. 289-298, Dec. 1993.

[2] J-C. Bolot and A. Vega Garcia "The case for FEC-based error control for packet audio in the Internet" *to appear in ACM Multimedia Systems*.

[3] R. Caceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics" *to appear in IEEE Trans. of Information Theory*.

[4] R. Caceres, N.G. Duffield, J .Horowitz, D. Towsley and T. Bu, "Multicast-Based Inference of Network Internal Loss Characteristics: Accuracy of Packet Estimation" *Proc. of Infocom '99*, New York, NY, Mar. 1999.

[5] R. Caceres, N.G. Duffield, J .Horowitz, F. Lo Presti and D. Towsley, "Loss-Based Inference of Multicast Network Topology" IEEE Conference on Decision and Control, 1999, to appear.

[6] CAIDA: Cooperative Association for Internet Data Analysis. For more information see http://www.caida.org

[7] K. Claffy, G. Polyzos and H-W. Braun, "Measurements Considerations for Assessing Unidirectional Latencies", *Interntworking: Research and Experience*, Vol. 4, no. 3, pp. 121-132, Sept. 1993.

[8] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE '96*, Oct. 1996.

[9] A. Downey, "Using pathchar to estimate Internet link characteristics, On Proceedings ACM SIGCOMM'99, Cambridge, MA.

[10] Felix: Independent Monitoring for Network Survivability. For more information see ftp://ftp.bellcore.com/pub/mwg/felix/index.html

[11] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, no. 4, August 1993.

[12] IPMA: Internet Performance Measurement and Analysis. For more information see http://www.merit.edu/ipma

[13] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see ftp://ftp.ee.lbl.gov/pathchar

[14] F. Lo Presti and N.G. Duffield, "Multicast-Based Inference of Network-Internal Delay Distributions", Preprint AT&T Laboratories and University of Massachusetts.

[15] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, "Creating a Scalable Architecture for Internet Measurement," *to appear in Proc. INET '98*.

[16] M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, Montreal, June 1996.

[17] D. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", *RFC 1305*, Network Information Center, SRI International, Menlo Park, CA, Mar. 1992.

[18] S. Moon, P. Skelly and D. Towsley, "Estimation and Removal of Clock Skew from Network Delay Measurements" *Proc. of Infocom '99*, New York, NY, Mar. 1999.

[19] mtrace – Print multicast path from a source to a receiver. For more information see ftp://ftp.parc.xerox.com/pub/net-research/ipmulti

[20] R.J. Muirhead, "Aspects of Multivariate Statistical Theory", Wiley, New York, 1982.

[21] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load", *Internetworking: Research and Experience*, Vol. 5, pp. 163-205, Dec. 1994.

[22] ns – Network Simulator. For more information see http://www-mash.cs.berkeley.edu/ns/ns.html

[23] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. SIGCOMM '96*, Stanford, Aug. 1996.

[24] V. Paxson, "End-to-End Internet Packet Dynamics," *Proc. SIGCOMM 1997*, Cannes, France, pp. 139-152, Sept. 1997.

[25] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. Dissertation, University of California, Berkeley, Apr. 1997.

[26] V. Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. SIGCOMM 1997*, Cannes, France, 167–179, Sept. 1997.

[27] V. Paxson, "On calibrating measurements of Packet Transit Times", *Proc. of SIGMETRICS '98*, Madison, June 1998.

[28] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", Proceedings IEEE Infocom' 99, New York, NY, Mar. 1999.

[29] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[30] Surveyor. For more information see http://io.advanced.org/surveyor/

Multicast Topology Inference from Measured End-to-End Loss

N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley

Abstract—**The use of multicast inference on end-to-end measurement has recently been proposed as a means to infer network internal characteristics such as packet link loss rate and delay. In this paper we propose three types of algorithm that use loss measurements to infer the underlying multicast topology: (i) a grouping estimator that exploits the monotonicity of loss rates with increasing path length; (ii) a maximum likelihood estimator; and (iii) a Bayesian estimator. We establish their consistency, compare their complexity and accuracy, and analyze the modes of failure and their asymptotic probabilities.**

Keywords: Communication Networks, End-to-End Measurement, Maximum Likelihood Estimation, Multicast, Statistical Inference, Topology Discovery.

I. Introduction

A. Motivation.

In this paper we propose and evaluate a number of algorithms for the inference of logical multicast topologies from end-to-end network measurements. All are developed from recent work that shows how to infer per link loss rate from measured end-to-end loss of multicast traffic. The idea behind this approach is that performance characteristics across a number of intersecting network paths can be combined to reveal characteristics of the intersection of the paths. In this way, one can infer characteristics across a portion of the path without requiring the portion's endpoints to terminate measurements.

The use of active multicast probes to perform measurements is particularly well suited to this approach due to the inherent correlations in packet loss seen at different receivers. Consider a multicast routing tree connecting the probe source to a number of receivers. When a probe packet is dispatched down the tree from the source, a copy is sent down each descendant link from every branch point encountered. By this action, one packet at the source gives rise to a copy of the packet at each receiver. Thus a packet reaching each member of a subset of receivers encounters *identical* conditions between the source and the receivers' closest common branch point in the tree.

N.G. Duffield and F. Lo Presti are with AT&T Labs–Research, 180 Park Avenue, Florham Park, NJ 07932, USA, E-mail: {duffield, lopresti}@research.att.com. J. Horowitz is with the Dept. of Math. & Statistics, University of Massachusetts, Amherst, MA 01003, USA, E-mail: joeh@math.umass.edu. D. Towsley is with the Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003, USA, E-mail: towsley@cs.umass.edu

This approach has been used to infer the per link packet loss probabilities for logical multicast trees with a known topology. The Maximum Likelihood Estimator (MLE) for the link probabilities was determined in [3] under the assumption that probe loss occurs independently across links and between probes. This estimate is somewhat robust with respect to violations of this assumption. This approach will be discussed in more detail presently.

The focus of the current paper is the extension of these methods to infer the *logical topology* when it is not known in advance. This is motivated in part by ongoing work [1] to incorporate the loss-based MLE into the National Internet Measurement Infrastructure [14]. In this case, inference is performed on end-to-end measurements arising from the exchange of multicast probes between a number of measurement hosts stationed in the Internet. The methods here can be used to infer first the logical multicast topology, and then the loss rates on the links in this topology. What we do not provide (are unable to) is an algorithm for identifying the physical topology of a network.

A more important motivation for this work is that knowledge of the multicast topology can be used by multicast applications. It has been shown in [9] that organizing a set of receivers in a bulk transfer application into a tree can substantially improve performance. Such an organization is central component of the widely used RMTP-II protocol [20]. The development of tree construction algorithms for the purpose of supporting reliable multicast has been identified to be of fundamental importance by the Reliable Multicast Transport Group of the IETF; see [7]. This motivated the work reported in [16], which was concerned with grouping multicast receivers that share the same set of network bottlenecks from the source for the purposes of congestion control. Closely related to [3], the approach of [16] is based on estimating packet loss rates for the path between the source and the common ancestor of pairs of nodes in the special case of binary trees. Since loss is a non-decreasing function of the path length, this quantity should be maximal for a sibling pair. The whole binary tree is reconstructed by iterating this procedure.

B. Contribution.

This paper describes and evaluates three methods for inference of logical multicast topology from end-to-end multicast measurements. Two of these ((i) and (ii) below) are directly based on the MLE for link loss probabilities of [3], as recounted in Section II. In more detail, the three methods are:

(i) Grouping Classifiers. We extend the grouping method of [16] to general trees, and establish its correctness. This is done in two steps. First, in Section III, we apply and extend the methods of [3] to establish a one-to-one correspondence between the the expected distribution of events measurable at the leaves, and the underlying topology and loss rates. In particular, we provide an algorithm that reconstructs arbitrary (e.g. non-binary) topologies from the corresponding distributions of leaf-measurable events. Second, in Section IV, we adapt the algorithm to work with the empirical leaf-event distributions arising from multicast end-to-end measurements. A complication arises through the fact that certain equalities that hold for the expected distributions only hold approximately for the measured distributions. We propose and evaluate three variants of the algorithm to overcome this. One is based on the above reconstruction method for general trees; the other two methods use binary grouping operations to reconstruct a binary tree, which is then manipulated to yield the inferred tree.

(ii) Maximum Likelihood Classifier. Given the measured end-to-end packet losses, the link loss estimator of [3] associates a likelihood with each possible logical multicast tree connecting the source to the receivers. The maximum likelihood classifier selects that tree for which the likelihood is maximal. This estimator is presented in Section V.

(iii) Bayesian Classifier. In this approach, the topology and link probabilities are treated as random variables with some prior distribution. In Bayesian decision theory one specifies a loss function that characterizes a penalty for misclassification, then selects the topology that minimizes the mean value of this penalty according to the posterior distribution (i.e. the conditional distribution of the parameters given the measurements). This estimator is presented in Section VI.

In all cases we establish that the classifiers are consistent, i.e., the probability of correct classification converges to 1 as the number of probes grows to infinity. We establish connections amongst the grouping-based algorithms. In particular, the general grouping-based algorithm is equivalent to the composition of the binary grouping algorithm with a pruning operation that excises links of zero loss and identifies their endpoints. The latter approach turns out to be computationally simpler.

The ML and Bayesian classifiers, embodying standard statistical methods, provide reference points for the accuracy of the grouping-based classifiers. In Section VII we use simulations to evaluate the relative accuracy of the topology classifiers, and to understand their modes of failure. We find that the accuracy of the grouping classifiers either closely matches or exceeds that of the other methods when applied to the identification of a selection of fixed unknown topologies. This finding is supported by some numerical results on the tail asymptotics of misclassification probabilities when using large numbers of probes. The simulations show the techniques can resolve topologies even when link loss probabilities are as small as about 1%, on the basis of data from a few thousand probes. This data could be gathered from a probe source of low bandwidth (a few tens of kbits per second) over a few minutes.

The ML and Bayesian classifiers are considerably more computationally complex than the grouping-based methods. This is for two reasons: (i) they exhaustively search the set of possible trees, whereas the grouping approaches progressively exclude certain topologies from consideration as groups are formed; (ii) their per-topology computational costs are greater. Since the number of possible topologies grows rapidly with the number of receivers, any decrease in per-topology cost for the ML and Bayesian classifiers would eventually be swamped by the growth in the number of possible topologies. For this reason, we expect significant decrease in complexity will only be available for classifiers that are able to search the topology space in a relatively sophisticated manner, e.g. as performed by the grouping-based algorithms. Summarizing, we conclude that binary-based grouping algorithms provide the best combination of accuracy and computational simplicity.

In Section VIII we further analyze the modes of misclassification in grouping algorithms. We distinguish the coarser notion of misgrouping, which entails failure to identify the descendant leaves of a given node. This notion is relevant, for example, in multicast congestion control, where one is interested in establishing the set of receivers that are behind each bottleneck. We obtain rates of convergence of the probability of successful grouping and classification in the regime of small link loss rates.

We conclude in Section IX; the proofs and some more detailed technical material are deferred to Section X.

C. Other Related Work.

The `mtrace` [12] measurement tool, reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss statistics. The `tracer` tool [10] uses `mtrace` to perform topology discovery. We briefly contrast some properties of those methods with those presented here. (i) Access: `mtrace` relies on routers to respond to explicit measurement queries; access to such facilities may be restricted

by service providers. The present method does not require such cooperation. (ii) Scaling: `mtrace` needs to run once per receiver in order to cover the tree, so that each router must process requests from all its descendant leaf nodes. The present method works with a single pass down the tree. On the other hand, our methods do not associate physical network addresses with nodes of the logical multicast tree. For this reason, we envisage combining `mtrace` and multicast-based estimation in measurement infrastructures, complementing infrequent `mtrace` measurements with ongoing multicast based-inference to detect topology changes.

In the broader context of network tomography we mention some recent analytic work on a different problem, namely, determination of source-destination traffic matrix from source- and destination-averaged traffic volumes; see [18], [19] for further details.

II. Loss Trees and Inference of Loss Rate

We begin by reviewing the tree and loss models used to formulate the MLE for link loss probabilities in a known topology. We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links that join them. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree has at least two children, except the leaf nodes (which have none) and the root (which we assume has one). We can construct the logical tree from the physical tree by the following procedure: except for the root, delete each node that has only one child, and adjust the link set accordingly by linking its parent directly to its child.

A. Tree Model.

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify one node, the root 0, with the source of probes, and set of leaves $R \subset V$ with the set of receivers. We say that a link is internal if neither of its endpoints is the root or a leaf node. We will occasionally use W to denote $V \setminus (\{0,1\} \cup R)$, where 1 denotes the child node of 0, the set of nodes terminating internal links. Each node k, apart from the root, has a parent $f(k)$ such that $(f(k), k) \in L$. We will sometimes refer to $(f(k), k)$ as link k. Define recursively the compositions $f^n = f \circ f^{n-1}$ with $f^1 = f$. We say j is descended from k, and write $j \prec k$, if $k = f^n(j)$ for some positive integer n. The set of children of k, namely $\{j \in V : f(j) = k\}$ is denoted by $d(k)$. The (nearest) ancestor $a(U)$ of a subset $U \subset V$ is the $\prec$-least upper bound of all the elements of U. A collection of nodes U are said to be siblings if they have the same parent, i.e., if $f(k) = a(U)\ \forall k \in U$. A maximal sibling set comprises the entire set $d(k)$ of children of some node $k \in V$. $\mathcal{T}(k) = (V(k), L(k))$ will denote the subtree rooted at k; $R(k) = R \cap V(k)$ is the set of receivers in $\mathcal{T}(k)$.

B. Loss Model.

For each link we assume an independent Bernoulli loss model: each probe is successfully transmitted across link k with probability α_k. Thus the progress of each probe down the tree is described by an independent copy of a stochastic process $X = (X_k)_{k \in V}$ as follows. $X_0 = 1$. $X_k = 1$ if the probe reaches node $k \in V$ and 0 otherwise. If $X_k = 0$, then $X_j = 0, \forall j \in d(k)$. Otherwise, $P[X_j = 1 | X_k = 1] = \alpha_j$ and $P[X_j = 0 | X_k = 1] = 1 - \alpha_j$. We adopt the convention $\alpha_0 = 1$ and denote $\alpha = (\alpha_i)_{i \in V}$. We call the pair $(\mathcal{T}, \alpha)$ a **loss tree**. $\mathsf{P}_{\mathcal{T},\alpha}$ will denote the distribution of X on the loss tree $(\mathcal{T}, \alpha)$. In what follows we shall work exclusively with **canonical loss trees**. A loss tree is said to be in canonical form if $0 < \alpha_k < 1, \forall k \in V$ except for $k = 0$. Any tree $(\mathcal{T}, \alpha)$ not in canonical form can be reduced to a loss tree, $(\mathcal{T}', \alpha')$, in canonical form such that the distribution of $(X_k)_{k \in R}$ is the same under the corresponding probabilities $\mathsf{P}_{\mathcal{T},\alpha}$ and $\mathsf{P}_{\mathcal{T}',\alpha'}$. To achieve this, links k with $\alpha_k = 1$ are excised and their endpoints identified. If any link k has $\alpha_k = 0$, then $X_j = 0$ for all $j \prec k$, and hence no probes are received at any receiver in $R(k)$. By removal of subtrees $\mathcal{T}(k)$ rooted at such k, we obtain a tree in which all probabilities $\alpha_k > 0$. Henceforth we shall consider only canonical loss trees.

C. Inference of Loss Rates.

When a probe is sent down the tree from the root 0, we can not observe the whole process X, but only the outcome $(X_k)_{k \in R} \in \Omega = \{0,1\}^R$ that indicates whether or not the probe reached each receiver. In [3] it was shown how the link probabilities can be determined from the the distribution of outcomes when the topology is known. Set

$$\gamma(k) = \mathsf{P}_{\mathcal{T},\alpha}[\vee_{j \in R(k)} X_j = 1]. \tag{1}$$

The internal link probabilities α can be found from $\gamma = \{\gamma(k) : k \in V\}$ as follows. For $k \in V$ let $A(k)$ be the probability that the probe reaches k. Thus $A(k) = \prod_{j \succeq k} \alpha_j$, the product of the probabilities of successful transmission on each link between k and the root 0. For $U \subset V$ we write $\gamma(U) = \mathsf{P}[\vee_{k \in U} \vee_{j \in R(k)} X_j = 1]$. A short probabilistic argument shows that for any $U \subseteq d(k)$,

$$(1 - \gamma(U)/A(k)) = \prod_{j \in U} (1 - \gamma(j)/A(k)). \tag{2}$$

In particular, this holds for $U = d(k)$ in which case $\gamma(U) = \gamma(k)$. It can be shown for canonical loss trees that $A(k)$ is the unique solution of (2); see Lemma 1 in [3] or Prop 1 below. Thus given $\{\gamma(k) : k \in V\}$ one

can find $(A(k))_{k\in V}$, and hence α, by taking appropriate quotients.

Let $x = (x^{(1)}, \ldots, x^{(n)})$ with $x^{(m)} = (X_k^{(m)})_{k\in R}$ be the set of outcomes arising from the dispatch of n probes from the source. We denote the log-likelihood function of this event by

$$\mathcal{L}(\mathcal{T}, \alpha) = \log \mathsf{P}_{\mathcal{T},\alpha}[x] \tag{3}$$

Construct the empirical distributions $\widehat{\gamma}(k) = n^{-1}\sum_{m=1}^{n} \vee_{j\in R(k)} X_j^{(m)}$, i.e. the fraction of the n probes that reaches some receiver descended from k. Let $\widehat{A}$ denote the corresponding solution of (2) obtained by using $\widehat{\gamma}$ in place of γ, and $\widehat{\alpha}$ the corresponding probabilities obtained by taking quotients of the $\widehat{A}$. The following results, the proof of which can be found in [3], holds.

Theorem 1: Let $\mathcal{T}$ be a canonical loss tree.
(i) The loss model is identifiable, i.e. $P_{\mathcal{T},\alpha} = P_{\mathcal{T},\alpha'}$ implies $\alpha = \alpha'$.
(ii) with probability 1, for sufficiently large n, $\widehat{A}, \widehat{\alpha}$ are the Maximum Likelihood Estimators of A, α, i.e.,

$$\widehat{\alpha} = \arg\max_{\alpha} \mathcal{L}(\mathcal{T}, \alpha). \tag{4}$$

As a consequence of the MLE property, $\widehat{A}$ is consistent ($\widehat{A} \overset{n\to\infty}{\longrightarrow} A$ with probability 1), and asymptotically normal ($\sqrt{n}(\widehat{A} - A)$ converges in distribution to a multivariate Gaussian random variable as $n \to \infty$), and similarly for α; see [17].

III. Deterministic Reconstruction of Loss Trees by Grouping

The use of estimates of shared loss rates at multicast receivers has been proposed recently in order to group multicast receivers that share the same set of bottlenecks on the path from the source [16]. The approach was formulated for binary trees, with shared loss rates having the direct interpretation of the loss rate on the path from the root to the (nearest) ancestor of two receivers. Since the loss rate cannot decrease as the path is extended, the pair of receivers for which shared loss rate is greatest will be siblings; if not then one of the receivers would have a sibling and the shared loss rate on the path to their ancestor would be greater. This maximizing pair is identified as a pair of siblings and replaced by a composite node that represents their parent. Iterating this procedure should then reconstruct the binary tree.

In this section and the following section we establish theoretically the correctness of this approach, and extend it to cover general trees, i.e., those with nodes whose branching ratio may be greater than two. In this section we describe how canonical loss trees are in one-to-one correspondence with the probability distributions of the random variables $(X_k)_{k\in R}$ visible at the receivers.

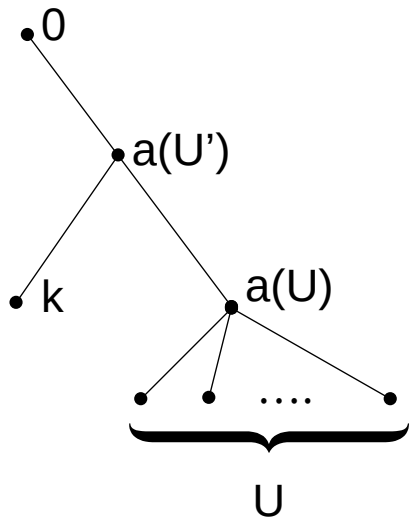


Fig. 1. $B(U') > B(U)$ where $U' = U \cup \{k\}$. Adjoining the non-sibling node k to U increases the value of B; see Prop. 1(iv).

Thus the loss tree can be recovered from the receiver probabilities. This is achieved by employing an analog of the shared loss for binary trees. This is a function $B(U)$ of the loss distribution at a set of nodes U that is minimized when U is a set of siblings, in which case $B(U) = A(a(U)))$, i.e. the complement of the shared loss rate to the nodes U. In the case of binary trees, we can identify the minimizing set U as siblings and substitute a composite node that represents their parent. Iterating this procedure should then reconstruct the tree. The definition and relevant properties of the function B are given in the following proposition.

Proposition 1: Let $\mathcal{T} = (V, L)$ be a canonical loss tree, and let $U \subset V$ with $\#U > 1$.
(i) The equation $(1 - \gamma(U)/B) = \prod_{k\in U}(1 - \gamma(k)/B)$ has a unique solution $B(U) > \gamma(U)$.
(ii) Let $B > \gamma(U)$. Then $(1 - \gamma(U)/B) > \prod_{k\in U}(1 - \gamma(k)/B)$ iff $B > B(U)$.
(iii) $B(U) = A(a(U))$ if U is a set of siblings, and hence $B(U)$ takes the same value for any sibling set with a given parent.
(iv) Let U be a set of siblings, and suppose $k \in V$ is such that $a(U \cup \{k\}) \succ a(U)$ and $a(U \cup \{k\}) \succ k$. Then $B(U \cup \{k\}) > B(U)$.

Proposition 1(iv) shows that adjoining a non-sibling non-ancestor node to a set of siblings can only increase the value of B; see Figure 1. This provides the means to reconstruct the tree $\mathcal{T}$ directly from the $\{\gamma(U) : U \subset R\}$. We call the procedure to do this the Deterministic Loss Tree Classification Algorithm (DLT), specified in Fig-

1. *Input:* The set of receivers R and associated probabilities $\{\gamma(U) : U \subset R\}$;
2. $R' := R$; $V' := R$; $L' := \emptyset$;
3. **foreach** $j \in R'$ **do** $B(j) := \gamma(j)$; **enddo**
4. **while** $|R'| > 1$ **do**
5. **select** $U = \{u_1, u_2\} \subseteq R'$ that minimizes $B(U)$;
6. **while** there exists $u \in R' \setminus U$ such that $B(U \cup \{u\}) = B(U)$ **do**
7. $U := U \cup \{u\}$;
8. **enddo**
9. $V' := V' \cup \{U\}$; $R' := (R' \setminus U) \cup \{U\}$;
10. **foreach** $j \in U$ **do**
11. $L' := L' \cup \{(U, j)\}$; $\alpha'(j) := B(j)/B(U)$;
12. **enddo**
13. **enddo**
14. $V' = V' \cup \{0\}$;
15. $L' = L' \cup \{0, U\}$;
16. $\alpha'_U = B(U)$; $\alpha'_0 = 1$;
17. *Output:* loss tree $((V', L'), \alpha')$.

Fig. 2. Deterministic Loss Tree Classification Algorithm (DLT).

1. *Input*: a loss tree $(\mathcal{T}, \alpha)$;
2. *Parameter*: a threshold $\varepsilon \geq 0$;
3. $V' := \{0\} \cup d_{\mathcal{T}}(0)$; $L' := \{(0, k) : k \in d_{\mathcal{T}}(0)\}$;
4. $U := d_{\mathcal{T}}(0)$;
5. **while** $U \neq \emptyset$ **do**
6. **select** $j \in U$;
7. $U := U \setminus \{j\} \cup d_{\mathcal{T}}(j)$;
8. **if** $((1 - \alpha_j) \leq \varepsilon) \wedge (j \neq R)$ **then**
9. $L' := (L' \cup \{(f_{\mathcal{T}'}(j), k) : k \in d_{\mathcal{T}}(j)\}) \setminus \{(f_{\mathcal{T}'}(j), j)\}$;
10. $V' := V' \setminus \{j\} \cup d_{\mathcal{T}}(j)$;
11. **else**
12. $L' := L' \cup \{(j, k) : k \in d_{\mathcal{T}}(j)\}$;
13. $V' := V' \cup d_{\mathcal{T}}(j)$;
14. **endif**;
15. **enddo**
16. *Output:* $((V', L'), \alpha')$

Fig. 3. Tree Pruning Algorithm TP(ε)

ure 2; it works as follows. At the start of each while loop from line 4, the set R' comprises those nodes available for grouping. We first find the pair $U = \{u_1, u_2\}$ that minimizes $B(U)$ (line 5), then progressively adjoin to it further elements that do not increase the value of B (lines 6 and 7). The members of the largest set obtained this way are identified as siblings; they are removed from the pool of nodes and replaced by their parent, designated by their union U (line 9). Links connecting U to its children (i.e. members) are added to the tree, and the link loss probabilities are determined by taking appropriate quotients of $B's$ (line 11). This process is repeated until all sibling sets have been identified. Finally, we adjoin the root node 0 and the link joining it to its single child (line 14).

Theorem 2: (i) DLT reconstructs any canonical loss tree $(\mathcal{T}, \alpha)$ from its receiver set R and the associated probabilities $\{\gamma(U) : U \subset R\}$.

(ii) Canonical loss trees are identifiable, i.e. $P_{\mathcal{T},\alpha} = P_{\mathcal{T}',\alpha'}$ implies that $(\mathcal{T}, \alpha) = (\mathcal{T}', \alpha')$.

Although we have not shown it here, it is possible to establish that any set R' present at line 4 of DLT has the property that $\min_{U \subset R'} B(U)$ is achieved when U is a sibling set. Consequently one could replace steps 5–8 of DLT by simply finding the maximal sibling set, i.e. select a maximal $U \subset R'$ that minimizes $B(U)$. However, this approach would have worse computational properties since it requires inspecting every subset of R'.

$B(U)$ is a root of the polynomial of degree $\#U - 1$ from Prop. 1(i). For a binary subset, $B(\{j, k\})$ is written down explicitly

$$B(\{j, k\}) = \frac{\gamma(j)\gamma(k)}{\gamma(k) + \gamma(j) - \gamma(\{j, k\})}; \qquad (5)$$

Calculation of $B(U)$ requires numerical root finding when $\#U > 5$. However, it is possible to recover $\mathcal{T}$ in a two stage procedure that requires the calculation of $B(U)$ only on binary sets U. The first stage uses the Deterministic Binary Loss Tree (DBLT) Classification Algorithm. DBLT is identical to DLT except that grouping is performed only over binary trees, thus omitting lines 6–8 in Figure 2. The second stage is to use a Tree Pruning (TP) Algorithm on the output of the DBLT. TP acts on a loss tree $((V, L), \alpha)$ by removing from L each internal link $(f(k), k)$ with loss rate $1 - \alpha_k = 0$ and identifying its endpoints $k, f(k)$. We will find it useful to specify a slightly more general version: for $\varepsilon \geq 0$, TP(ε) prunes link k when $1 - \alpha_k \leq \varepsilon$. We formally specify TP(ε) in Figure 3. In Section X we prove that composing the binary algorithm DBLT with pruning recovers the same topology as DLT for general canonical loss trees:

Theorem 3: DLT=TP(0)$\circ$DBLT for canonical loss trees.

IV. Inference of Loss Tree from Measured Leaf Probabilities

In this section, we present algorithms which adapt DLT to use the measured probabilities corresponding to the γ. Let $(X_k^{(i)})_{k \in R}^{i=1,\ldots,n}$ denote the measured outcomes arising from each of n probes. Define the processes $Y_k^{(i)}$ recur-

1. *Input*: The set of receivers R, number of probes n, receiver traces $(X_k^{(i)})_{k\in R}^{i=1,2,\ldots n}$;
2. $R' := R, V' := R;\ L' = \emptyset$;
3. **foreach** $k \in R$, **do**
4. $\quad \widehat{B}(k) := n^{-1}\sum_{i=1}^n X_k^{(i)}$;
5. $\quad$ **foreach** $i = 1,\ldots,n$, **do** $Y_k^{(i)} = X_k^{(i)}$; **enddo** ;
6. **enddo**
7. **while** $|R'| > 1$ **do**
8. $\quad$ **select** $U = \{u_1, u_2\} \subset R'$ that minimizes $\widehat{B}(U) = \frac{\sum_{i=1}^n Y_{u_1}^{(i)} \sum_{i=1}^n Y_{u_2}^{(i)}}{n\sum_{i=1}^n Y_{u_1}^{(i)} Y_{u_2}^{(i)}}$;
9. $\quad$ **foreach** $i = 1,\ldots,n$ **do** $Y_U^{(i)} = \vee_{u\in U} Y_u^{(i)}$ **enddo**
10. $\quad V := V \cup \{U\}$; $R' := (R' \setminus U) \cup \{U\}$;
11. $\quad$ **foreach** $u \in U$ **do**
12. $\quad\quad L' := L' \cup \{(U,u)\}$; $\widehat{\alpha}_u := \widehat{B}(u)/\widehat{B}(U)$;
13. $\quad$ **enddo**
14. **enddo**
15. $V' = V' \cup \{0\}$; $L' = L' \cup \{0, U\}$;
16. $\widehat{\alpha}_U = \widehat{B}(U)$; $\widehat{\alpha}_0 = 1$;
17. *Output:* loss tree $((V', L'), \widehat{\alpha})$.

Fig. 4. Binary Loss Tree Classification Algorithm (BLT).

sively by

$$Y_k^{(i)} = \vee_{j\in d(k)} Y_j^{(i)} \quad \text{with} \quad Y_k^{(i)} = X_k^{(i)},\ k \in R. \quad (6)$$

Thus $Y_k^i = 1$ iff probe i was received at some receiver descended from k; $\widehat{\gamma}(k) = n^{-1}\sum_{i=1}^n Y_k^{(i)}$ is the fraction of the probes $1,\ldots,n$ that reach some receiver descended from k. For $U \subset V$ we define $\widehat{\gamma}(U) = n^{-1}\sum_{i=1}^n \vee_{j\in U} Y_j^{(i)}$ analogously; $\widehat{\gamma}(U)$ is the fraction of probes that reach some receiver descended from some node in U. Let $\widehat{B}(U)$ be the unique solution in Prop. 1(ii) obtained by using $\widehat{\gamma}$ in place of γ. We will use the notation $(\widehat{\mathcal{T}}, \widehat{\alpha})$ to denote an inferred loss tree; sometimes we will use $\widehat{\mathcal{T}}_{\mathsf{X}}$ to distinguish the topology inferred by a particular algorithm X. P_{X}^f will denote the probability of false identification of topology $\mathcal{T}$ of the loss tree $(\mathcal{T}, \alpha)$ i.e. $P_{\mathsf{X}}^f = \mathsf{P}_{\mathcal{T},\alpha}[\widehat{\mathcal{T}}_{\mathsf{X}} \neq \mathcal{T}]$.

Theorem 4: Let $((V,L),\alpha)$ be a loss tree. Then $\lim_{n\to\infty} \widehat{B}(U) = B(U)$ for each $U \subset V$.

A. Classification of Binary Loss Trees

The adaptation of DLT is most straightforward for binary trees. By using $\widehat{B}$ in place of B in DLT and restricting the minimization of $\widehat{B}$ to binary sets we obtain the Binary Loss Tree (BLT) Classification Algorithm; we specify it formally in Figure 4. This is, essentially, the algorithm proposed in [16]. We have taken advantage of the recursive structure of the $Y_k^{(i)}$ (in line 9) in order to calculate the probabilities $\widehat{\gamma}$. Note that when BLT reconstructs an incorrect topology $\widehat{\mathcal{T}} \neq \mathcal{T}$, the definitions of quantities such as $\widehat{B}(U)$ and $Y_U^{(i)}$ extend evidently to subsets U of nodes in the incorrect topology $\mathcal{T}'$. The following theorem establishes the consistency of the estimator $\widehat{\mathcal{T}}_{\mathsf{BLT}}$; the proof appears in Section X.

Theorem 5: Let $(\mathcal{T},\alpha)$ be a binary canonical loss tree. With probability 1, $\widehat{\mathcal{T}}_{\mathsf{BLT}} = \mathcal{T}$ for all sufficiently large n, and hence $\lim_{n\to\infty} P_{\mathsf{BLT}}^f = 0$.

B. Classification of General Loss Trees

The adaptation of DLT to the classification of general loss trees from measured leaf probabilities is somewhat more complicated than the binary case. It is shown during the proof of Theorem 5 that the $\widehat{B}(U)$ have the same relative ordering as the $B(U)$ for n sufficiently large. But for a general tree (V,L), $B(U')$ takes the same value for any subset U' of a maximal sibling set $U \subset V$. For finitely many probes, the corresponding $\{\widehat{B}(U') : U' \subset U\}$ will not in general be equal. Hence choosing to group the subset U' that minimizes $\widehat{B}(\cdot)$ will not necessarily group all the siblings in U.

In this section we present three algorithms to classify general trees. Each of these overcomes the problem described in the previous paragraph by incorporating a threshold into the grouping procedure. The set U is grouped if $\widehat{B}(U)$ is sufficiently close to being minimal. However, this can also give rise to false inclusion by effectively ignoring internal links whose loss rates do not exceed the threshold. The variety of algorithms derives from different ways to implement the threshold. We establish domains in which the algorithms correctly classify canonical loss trees. In succeeding sections we evaluate their relative efficiencies and compare their modes and frequencies of false classification.

B.1 Binary Loss Tree Pruning Classification Algorithm BLTP.

Nodes are grouped as if the tree were binary, the resulting tree is pruned with TP(ε) to remove all internal links with loss probabilities less than or equal to the threshold $\varepsilon > 0$. Thus for each $\varepsilon > 0$ we define BLTP(ε) to be the composition TP(ε)$\circ$BLT. A refinement BLTP$'(\varepsilon)$ of BLTP(ε) is to recalculate the loss probabilities α' based on the measurements and the pruned topology $\mathcal{T}'$.

B.2 Binary Loss Tree Clique Classification Algorithm BLTC.

For each $\varepsilon > 0$, BLTC(ε) groups by forming maximal sets of nodes U in which all binary subsets U' have $\widehat{B}(U')$ sufficiently close to the true minimum over all binary sets. This amounts to replacing line 8 in Figure 4 with the following steps:

(i) select $U' = \{u', v'\}$ that minimizes $\widehat{B}(U')$;

(ii) construct the graph G of all links (u'', v'') such that $(1-\varepsilon)\widehat{B}(\{u'', v''\}) < \widehat{B}(U')$;
(iii) select U comprising the elements of the largest connected component of G that contains U'.
Note that if the grouping is done correctly, then $B(\{u', v'\})$ takes the same value for all binary subsets $\{u', v'\}$ of U. For finite but large n, the corresponding sampled $\widehat{B}(\{u', v'\})$ will differ slightly.

B.3 General Loss Tree Classification Algorithm GLT.

For each $\varepsilon > 0$, GLT(ε) is a modification of DLT that employs a threshold ε to perform the grouping based on $\widehat{B}$. Each grouping step starts by finding a binary set $\{u_1, u_2\}$ of minimal $\widehat{B}$, then adjoining further elements to it provided the resulting set U satisfies $\widehat{B}(U)(1-\varepsilon) < \widehat{B}(\{u_1, u_2\})$. The violation of this condition has the interpretation that the ancestor $a(U)$ is separated from $a(\{u_1, u_2\})$ by a link with loss rate at least ε. Thus we replace line 8 of Figure 4 by the following.

8a. **select** $U := \{u_1, u_2\} \subset S$ that minimizes $\widehat{B}(\cdot)$;
8b. **while** there exists $u \in R' \setminus U$ such that
$(1-\varepsilon)\widehat{B}(U \cup \{u\}) < \widehat{B}(u_1, u_2)\}$ **do**
8c. $U = U \cup \{u\}$;
8d. **enddo**

For clarity we have omitted the details of the dependence of $\widehat{B}$ on the $\widehat{\gamma}$; these are as described before Theorem 4.

B.4 Convergence of General Loss Tree Classifiers.

As the number of probes grows, the topology estimates furnished by BLTP(ε),BLTC(ε) and GLT(ε) converge to the true topology provided all internal link loss probabilities are greater than ε. This happens for the same reason as it does in BLT. It is not difficult to see that the deterministic versions of each algorithm, obtained by using B in place of $\widehat{B}$, reconstruct the topology. Since $\widehat{B}$ converges to B as the number of probes grows, the same is true for the classifiers using $\widehat{B}$. We collect these results without further proof:

Theorem 6: Let $(\mathcal{T}, \alpha)$ be a loss tree in which all loss probabilities $1 - \alpha_k > \varepsilon'$, $k \in W$, for some $\varepsilon' > 0$. For each $\varepsilon \in (0, \varepsilon')$ and each algorithm $\mathsf{X} \in \{\mathsf{BLTP}(\varepsilon), \mathsf{BLTC}(\varepsilon), \mathsf{GLT}(\varepsilon)\}$, with probability 1, $\widehat{\mathcal{T}}_{\mathsf{X}} = \mathcal{T}$ for all sufficiently large n, and hence $\lim_{n\to\infty} P^f_{\mathsf{X}} = 0$.

Convergence to the true topology requires ε to be smaller than the internal link loss rates, which are typically not known in advance. A very small value of ε is more likely to satisfy the above condition but at the cost, as shown in Section VIII, of slower classifier convergence. A large value of ε, on the other hand, is more likely to result in systematically removing links with small loss rates. In practice, however, we believe that the choice of ε does not pose a problem. We expect, indeed, that for many applications while it is important to correctly identify links with high loss rate, it could be considered acceptable failure to detect those with small loss rates. In other words, in practice, it could be sufficient the convergence of the inferred topology to $\mathcal{T}^\varepsilon = \mathsf{TP}(\varepsilon)(\mathcal{T})$ obtained from $\mathcal{T}$ by ignoring links whose loss rates fell below some specific value ε which, in this case, would be regarded as some application-specific minimum loss rate of interest.

The results below establish the desired convergence to $\mathcal{T}^\varepsilon$ for any $\varepsilon \in (0, 1)$ provided $\varepsilon \neq \alpha_k$, $k \in W$. The key observation is that since the deterministic versions of each algorithm reconstruct $\mathcal{T}^\varepsilon$, so does each algorithm, as the number of probes grows. Denote $P^f_X(\varepsilon) = \mathsf{P}_{\mathcal{T},\alpha}[\widehat{\mathcal{T}}_X \neq \mathcal{T}^\varepsilon]$. Without further proof we have:

Theorem 7: Let $(\mathcal{T}, \alpha)$ be a loss tree. For each $\varepsilon \in (0, 1)$, such that $\varepsilon \neq \alpha_k$, $k \in W$, and for each algorithm $\mathsf{X} \in \{\mathsf{BLTP}(\varepsilon), \mathsf{BLTC}(\varepsilon), \mathsf{GLT}(\varepsilon)\}$, then with probability 1, $\widehat{\mathcal{T}}_{\mathsf{X}} = \mathcal{T}^\varepsilon = \mathsf{TP}(\varepsilon)(\mathcal{T})$ for all sufficiently large n, and hence $\lim_{n\to\infty} P^f_{\mathsf{X}}(\varepsilon) = 0$.

C. Effects of Model Violation

The two underlying statistical assumptions are (i) probes are independent; and (ii) conditioned on a probe having reached a given node k, the events of probe loss on distinct subtrees descended from k are independent. We now discuss the impact of violations of these assumptions.

The first observation is that the estimators remain consistent under the introduction of some temporal dependence between probes, i.e. under violation of assumption (i) above. Assuming the loss process to be ergodic, $\widehat{\gamma}$ still converges to γ almost surely, as the number of probes n grows. However, rates of convergence can be slower, and hence the variance of $\widehat{B}$ can be higher, than for the independent case. This would increase the misclassification probabilities for inference from a given number of probes n.

On the other hand, spatial dependence of loss (i.e. violations of assumption (ii) above) can lead to bias. We take spatial loss dependence to be characterized by departure from zero of an appropriate set of loss correlation coefficients. By extending an argument given for binary trees in [3, Theorem 7], it can be shown that the limit quantities $B' = \lim_{n\to\infty} \widehat{B}$ deform continuously away from the quantities B of the spatially independent case as the loss correlation coefficients move away from zero. Hence a given canonical loss tree can be recovered correctly by applying DBLT to the quantities B' provided the spatial dependence is sufficiently small, i.e., to make the B' sufficiently close to B' so that $B(U_1) > B(U_2)$ iff $B'(U_1) > B'(U_2)$ for all relevant subsets of nodes U_1 and U_2. Then by a similar argument to that of Theorem 5,

a tree with link loss rates greater than some $\varepsilon > 0$, is recovered by BLTP(ε) with probability 1 for a sufficiently large number n of probes, and sufficiently small spatial correlations.

We remark that the the experiments reported in Sections VII and VIII use network level simulation rather than model based simulation. Hence it is expected that the model assumptions will be violated to some extent. Nevertheless, the classifiers are found to be quite accurate.

V. Maximum-Likelihood Classifier

Let $\mathcal{T}(R)$ denote the set of logical multicast trees with receiver set R. Denote by $\widehat{\alpha}_{\mathcal{T}}$ the MLE of α in (4) for the topology $\mathcal{T}$. The *maximum-likelihood (ML) classifier* assigns the topology $\widehat{\mathcal{T}}_{\rm ML}$ that maximizes $\mathcal{L}(\mathcal{T}, \widehat{\alpha}_{\mathcal{T}})$:

$$\widehat{\mathcal{T}}_{\rm ML} = \arg\max_{\mathcal{T} \in \mathcal{T}(R)} \mathcal{L}(\mathcal{T}, \widehat{\alpha}_{\mathcal{T}}). \tag{7}$$

We prove that, if the link probabilities are bounded away from 0 and 1, the ML-classifier is *consistent* in the sense that, w.p. 1, it identifies the correct topology as the number of probes grows to infinity. For $\varepsilon > 0$, let $\mathcal{A}^{\varepsilon}_{\mathcal{T}} = \{\alpha : \varepsilon \le \alpha_k \le 1-\varepsilon,\ k \in V \setminus \{0\}\}$.

Theorem 8: Let $\varepsilon > 0$ and let $(\mathcal{T}, \alpha)$ be a loss tree with $\alpha \in \mathcal{A}^{\varepsilon}_{\mathcal{T}}$. Then $\lim_{n\to\infty} \mathsf{P}_{\mathcal{T},\alpha}[\widehat{\mathcal{T}}_{\rm ML} \neq \mathcal{T}] = 0$.

VI. Loss-Based Bayesian Tree Classifier

Let $\mathcal{T}(R)$ denote the set of logical multicast topologies having a given receiver set R. $\mathcal{A}^0_{\mathcal{T}}$ from Section V, is the set of possible loss rates in the topology $\mathcal{T}$. A possible loss tree with topology in $\mathcal{T}(R)$ is an element of the parameter space

$$\Theta = \cup_{\tau \in \mathcal{T}(R)} (\{\tau\} \times \mathcal{A}^0_{\tau}). \tag{8}$$

Let $\pi(\tau, \alpha)$ be a prior distribution on Θ. Given receiver measurements $x = (x^{(1)}, \ldots, x^{(n)})$, the posterior distribution on Θ is

$$\pi(\tau, \alpha | x) = \pi(\tau, \alpha) f(x | \tau, \alpha) / f(x), \tag{9}$$

where $f(x|\tau,\alpha) = e^{\mathcal{L}(\tau,\alpha)}$ is the joint density of the observations and $f(x)$ their marginal density.

A *decision rule* δ provides an estimate $\delta(x) \in \Theta$ of the loss tree given receiver measurements x. The quality of a decision rule is evaluated in terms of a *loss function* $H(\theta, \theta')$, a nonnegative function on $\Theta \times \Theta$ interpreted as the loss incurred by deciding that θ' is the true parameter when, in fact, it is θ. A measure of quality of a decision rule δ is its Bayes risk $R(\delta) = E(H(\theta, \delta(x))$, where the expectation is taken with respect to the joint distribution $\pi(\tau,\alpha) f(x \mid \tau, \alpha)$. of the loss tree $\theta = (\tau, \alpha)$ and the observations x. The Bayes decision rule δ_B is the one that minimizes $R(\delta)$: it has least average loss. A standard theorem in decision theory gives δ_B in the form:

$$\delta_B(x) = \arg\min_{\theta' \in \Theta} \int_{\Theta} H(\theta, \theta') \pi(\theta | x)\, d\theta, \tag{10}$$

i.e., it is the minimizer of the *posterior risk*, which is the expected loss with respect to the posterior distribution $\pi(\theta|x)$; see Prop. 3.16 of [17] and Section 4.4, result 1 of [2].

Since our interest is in identifying the correct topology, we choose the loss function $H((\tau,\alpha),(\tau',\alpha')) = \chi[\tau \neq \tau']$ where χ is the indicator function, i.e., no loss for a correct identification of the topology, and unit loss for any misidentification. Here, the loss rates α play the role of a nuisance parameter. The Bayes classifier for the topology becomes $\widehat{\mathcal{T}}_B = \widehat{\mathcal{T}}_B(x)$, where

$$\widehat{\mathcal{T}}_B(x) = \arg\min_{\tau' \in \mathcal{T}(R)} \mathsf{P}[\tau' \neq \tau \mid x], \tag{11}$$

or, equivalently,

$$\widehat{\mathcal{T}}_B(x) = \arg\max_{\tau' \in \mathcal{T}(R)} \mathsf{P}[\tau' = \tau \mid x]. \tag{12}$$

Thus the Bayes classifier $\widehat{\mathcal{T}}_B$ yields the topology with maximum posterior probability given the data x. By definition, this classifier minimizes the misclassification probability.

A special case is the uniform prior in which all topologies in $\mathcal{T}(R)$ are taken to be equally likely, and for each topology τ, α is distributed uniformly on $\mathcal{A}^0_{\tau}$. The corresponding prior distribution, $\pi(\tau,\alpha) = \chi_{\mathcal{A}^0_{\tau}}(\alpha)/\#\mathcal{T}(R)$, is a non-informative prior, expressing "maximum ignorance" about the tree topology and link probabilities. Clearly if other prior information is available about the tree, it may be incorporated into a non-uniform prior distribution. The Bayes classifier becomes

$$\widehat{\mathcal{T}}_B(x) = \arg\max_{\tau' \in \mathcal{T}(R)} \int_{A^0_{\tau'}} f(x | \tau', \alpha)\, d\alpha. \tag{13}$$

This should be compared with the ML classifier in (7).

A. Consistency of Pseudo-Bayes Classifiers.

In practice our task is to identify the specific topology giving rise to a set of measured data. When no prior distribution is specified, the concept of the Bayes classifier, as the maximizer of the probability of correct classification, does not make sense, because "the" probability of correct classification is not defined. Nonetheless it may be convenient to construct a *pseudo-Bayes* classifier by choosing a distribution π on Θ, which plays the role of a prior, and forming the classifier in (10), which we now denote by

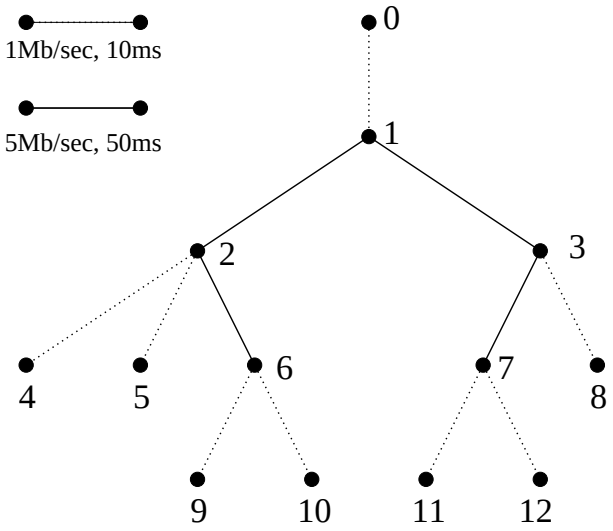


Fig. 5. Network-level simulation topology for ns. Links are of two types: *edge* links of 1Mb/s capacity and 10ms latency, and *interior* links of 5Mb/s capacity and 50ms latency.

$\widehat{\mathcal{T}}_\pi$. Classifiers constructed in this way are also consistent under a mild condition.

Theorem 9: Let π be a prior distribution on Θ, and assume that $(\mathcal{T}, \alpha)$ lies in the support of π. Then $\widehat{\mathcal{T}}_\pi$ is consistent in the frequentist sense, i.e., $\mathsf{P}_{\mathcal{T},\alpha}[\widehat{\mathcal{T}}_\pi \neq \mathcal{T}] \to 0$ as $n \to \infty$.

VII. Simulation Evaluation and Algorithm Comparison

A. Methodology.

We used two types of simulation to verify the accuracy of the classification algorithms and to compare their performance. In model-based simulation, packet loss occurs pseudorandomly in accordance with the independence assumptions of the model. This allows us to verify the prediction of the model in a controlled environment, and to rapidly investigate the performance of the classifiers in a varied set of topologies.

This approach was complemented by network-level simulations using the ns [13] program; these allow protocol-level simulation of probe traffic mixed in with background traffic of TCP and UDP sessions. Losses are due to buffer overflow, rather than being generated by a model, and hence can violate the Bernoulli assumptions underlying the analysis. This enables us to test the robustness to realistic violations of the model. For the ns simulations we used the topology shown in Figure 5. Links in the interior of the tree have higher capacity (5Mb/sec) and latency (50ms) than those at the edge (1Mb/sec and 10ms) in order to capture the heterogeneity between edges and core of a wide area network. Probes were generated from node 0 as a Poisson process with mean interarrival time 16ms. Background traffic comprised a mix of infinite FTP data sources connecting with TCP, and exponential on-off sources using UDP. The amount of background traffic was tuned in order to give link loss rates that could have significant performance impact on applications, down to as low as about 1%. One strength of our methodology is its ability to discern links with such small but potentially significant loss rates. In view of this, we will find it convenient to quote all loss rates as percentages.

B. Performance of Algorithms Based on Grouping

B.1 Dependence of Accuracy on Threshold ε.

We conducted 100 ns simulations of the three algorithms BLTP,BLTC and GLT. Link loss rates ranged from 1.8% to 10.9% on interior links; these are the links that must be resolved if the tree is to be correctly classified. In Figures 6–11 we plot the fraction of experiments in which the topology was correctly identified as a function of the number of probes, for the three algorithms, and for selected values of ε between 0.25% and 5%. Accuracy is best for intermediate ε, decreasing for larger and smaller ε. The explanation for this behavior is that smaller values of ε lead to stricter criteria for grouping nodes. With finitely many samples, for small ε, sufficiently large fluctuations of the $\widehat{B}$ cause erroneous exclusion of nodes. By increasing ε, the threshold for group formation is increased and so accuracy is initially increased. However, as ε approaches the smallest interior link loss rate, large fluctuations of the $\widehat{B}$ now cause erroneous inclusion of nodes into groups. When ε is moved much beyond the smallest interior loss rate, the probability of correct classification falls to zero. The behavior is different if we ignore failures to detect links with loss rates smaller than ε. For $\varepsilon = 5\%$ and $\varepsilon = 7\%$, in Figure 12 and 13, respectively, we plot the fraction of experiments in which the pruned topology $\mathcal{T}^\varepsilon$ was correctly identified for the three algorithms. Here the accuracy depends on the relative values of ε and the internal link loss rates. In these experiments, the actual loss rates was often very close to 5%, so that small fluctuations results in erroneous inclusions/exclusions of nodes which accounts for the significant fraction of failures for $\varepsilon = 5\%$. In Section VIII-B we shall analyze this behavior and obtain estimates for the probabilities of misclassification in the regimes described. We comment on the relative accuracy of the algorithms below.

B.2 Dependence of Accuracy on Topology.

We performed 1000 model-based simulations using randomly generated 24-node trees with given maximum branching ratios 2 and 4. Link loss rates were chosen at random in the interval $[1\%, 10\%]$. Figure 14 shows

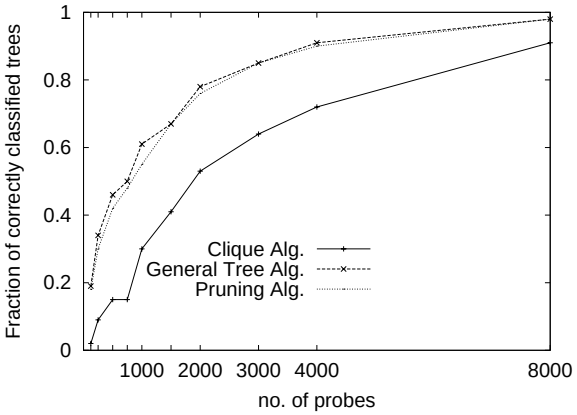


Fig. 6. $\epsilon = 0.25\%$.

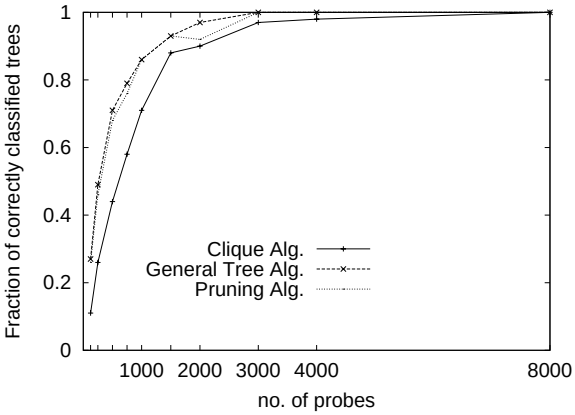


Fig. 7. $\epsilon = 0.5\%$.

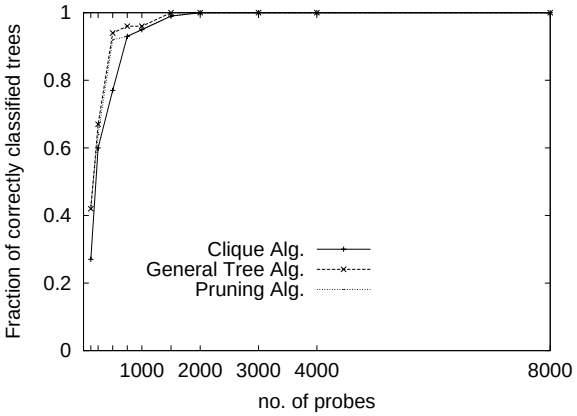


Fig. 8. $\epsilon = 1.0\%$.

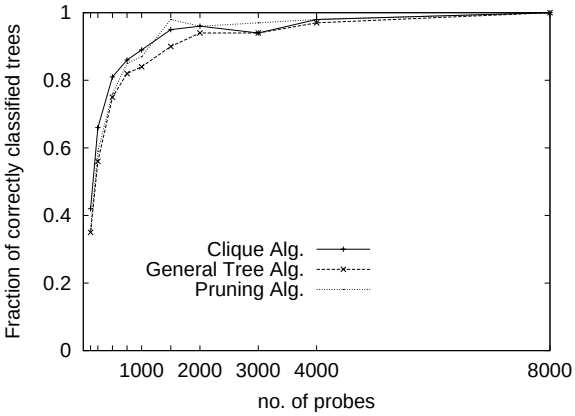


Fig. 9. $\epsilon = 2.0\%$.

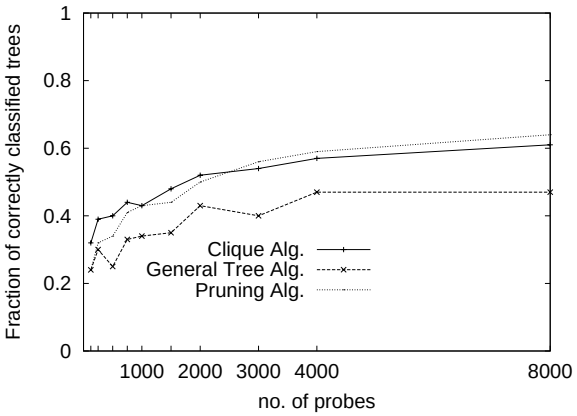


Fig. 10. $\epsilon = 3.0\%$.

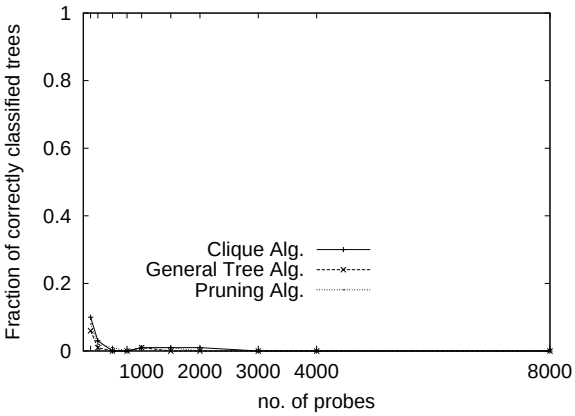


Fig. 11. $\epsilon = 5.0\%$.

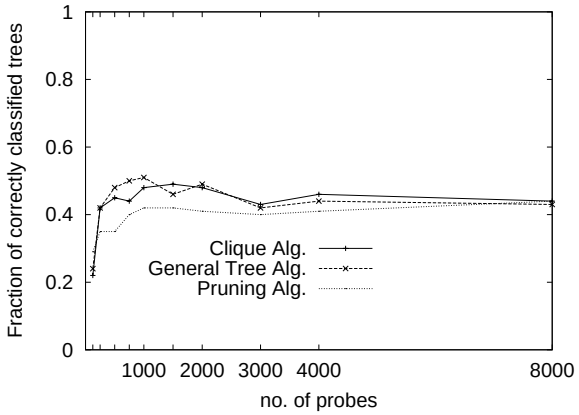


Fig. 12. $\epsilon = 5\%$.

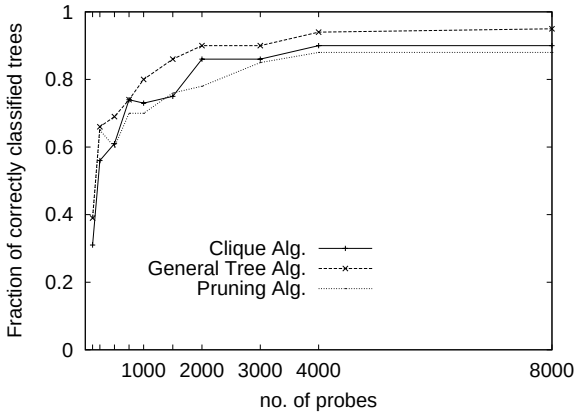


Fig. 13. $\epsilon = 7\%$.

the probability of successful classification for BLTP(ε), BLTC(ε) and GLT(ε) for $\varepsilon = 0.25\%$. In both cases this grows to 1, but convergence is slower for trees with higher branching ratios. We believe this behavior occurs due to the larger number of comparisons of values of $\widehat{B}$ that are made for trees with higher branching ratio, each such comparison affording an opportunity for misclassification.

B.3 Comparison of Grouping Algorithm Accuracy.

In all experiments reported so far, with one exception, the accuracies of BLTP and GLT were similar, and at least as good as that of BLTC. The similar behavior of BLTP and GLT is explained by observing that the two algorithms group nodes in a similar manner. In BLTP, a link is pruned from the reconstructed binary tree if its inferred loss rate is smaller than ε. In GLT, a node is added to a group if the estimated common loss of the augmented group is within ε of the estimated common loss of the original group. The operation of BLTC is somewhat different, checking all possible pairs amongst candidate nodes for grouping. Incorrect ordering in any test can result in false exclusion from a sibling set. We observe also that the performance gap between BLTC and the other algorithms is sensitive to the value of ε and to the branching ratio. The exceptional case in which BLTC performs better than the other algorithms is in the inference of binary trees: here BLTC performs slightly better because of the stricter grouping condition is employs, making it less likely to group more than two nodes.

B.4 Computational Complexity of Grouping Algorithms.

Of the two best performing grouping algorithms, namely BLTP and GLT, we observe that BLTP has smaller computational complexity for several reasons. First, $\widehat{B}$ is given explicitly for binary groups, whereas generally it requires numerical root finding. Second, although the algorithms have to calculate $\widehat{B}$ for up to $O(\#R^3)$ groups, in typical cases GLT requires additional calculations due to the larger sibling groups considered. Thirdly, observe that each increase of the size of sets considered in GLT is functionally equivalent to one pruning phase in BLTP. Thus in GLT, the threshold ε is applied throughout the algorithm; in BLTP it is applied only at the end. We expect this to facilitate adaptive selection of ε in BLTP. Comparing now with BLTC, we observe that this algorithm requires, in addition to the calculation of shared losses, the computation of a maximal connected subgraph, an operation that does not scale well for large numbers of nodes. For these reasons we adopt BLTP as our reference grouping algorithm since it is the simplest and has close to the best accuracy. In the next section, we compare its performance with that of the ML and Bayesian classifiers.

C. *Comparison of* BLTP *with the ML and Bayesian Classifiers*

C.1 Complexity.

In this section we compare our reference grouping algorithm, BLTP, with the ML and Bayesian classifiers. Here we consider the simplest implementation of these classifiers whereby we proceed by exhaustive search of the set $\mathcal{T}(R)$ of all possible topologies during evaluation of the maxima (7) and (13). By contrast, all the grouping algorithms proceed by eliminating subsets of $\mathcal{T}(R)$ from consideration; once a set of nodes is grouped, then only topologies which have those nodes as siblings are considered.

The Bayesian classifier further requires numerical integration for each candidate topology. In order to reduce its complexity we took the prior for the link rates to be uniform on the discrete set $\{1\%, \ldots, 10\%\}$, with all topolo-

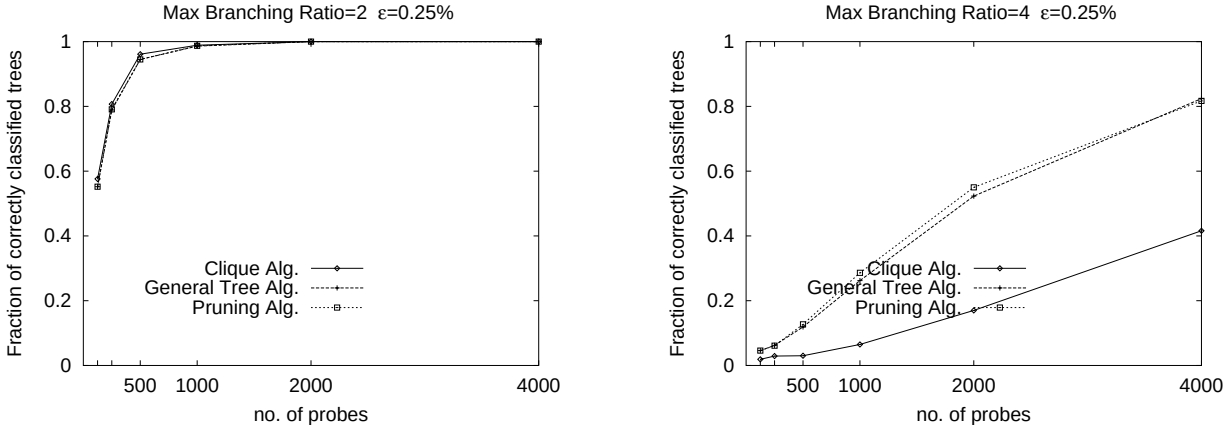


Fig. 14. DEPENDENCE OF ACCURACY ON BRANCHING RATIO: convergence is faster for binary trees (left); GLT and BLTP outperform BLTC for non-binary trees (right).

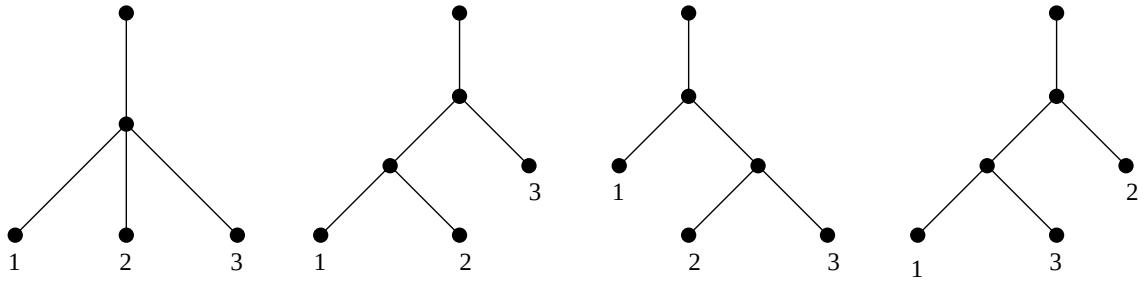


Fig. 15. ML AND BAYESIAN CLASSIFIER: The four possible topologies with three receivers.

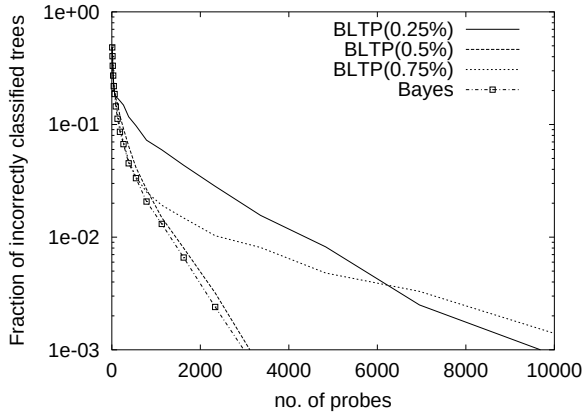
gies equally likely; we also precomputed the joint distributions $f(x|\tau,\alpha)$. Due to these computational costs, we were able to compare BLTP with the ML classifier for only up to five receivers, and restricted the Bayesian classifier to the smallest non-trivial case, that of three receivers. The four possible three-receiver trees are shown in Figure 15. In this case, the execution time of the Bayesian classifier was one order of magnitude longer than that of the ML classifier, and about two orders of magnitude longer than that of BLTP.

C.2 Relative Accuracy.

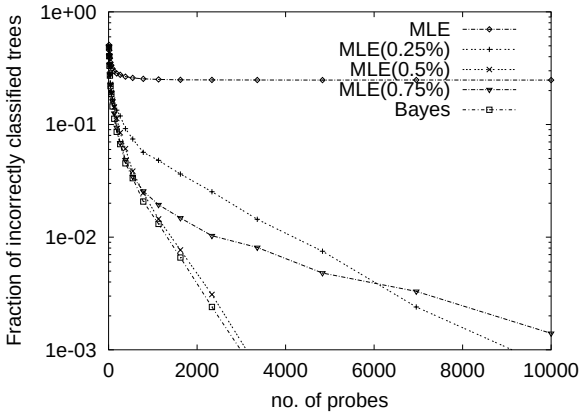
We conducted 10,000 simulations with the loss tree (τ,α) selected randomly according to the uniform prior. As remarked in Section VI, the Bayesian Classifier is, by definition, optimal in this setting. This is seen to be the case in Figure 16, where we plot the fraction of experiments in which the topology was incorrectly identified as function of the number of probes, for the different classifiers (for clarity we plot separately the curves for the ML and BLTP(ε) classifiers). Accuracy of BLTP greatly varies with ε: it gets close to optimal for the intermediate value of $\varepsilon = 0.5\%$, but rapidly decreases otherwise as ε approaches either 0 or the smallest internal link loss rate. It is interesting to observe that the ML classifier fails 25% of the time. This occurs when τ is the non-binary tree at the left in Figure 15. The reason is that the likelihood function is invariant under the insertion of links with zero loss. Statistical fluctuations present with finitely many probes lead to tree with highest likelihood to be a binary tree obtained by insertion of links with near-zero loss. This behavior does not contradict the consistency property of the ML classifier in Theorem 8; if links with loss less than some $\varepsilon > 0$ are excluded from consideration, then for sufficiently large number of probes, the spurious insertion of links will not occur.

The effect of these insertions can be suppressed by pruning after ML classification. Setting $\mathrm{ML}(\varepsilon) = \mathsf{TP}(\varepsilon) \circ \mathrm{ML}$ we find the accuracy almost identical with that of BLTP(ε); this is plotted in Figure 16(b). A more detailed inspection of the experiments shows that BLTP selects the maximum likelihood topology most of the time.

In practice we want to classify a fixed but unknown topology. In this context the uniform prior specifies a pseudo-Bayesian classifier, as in Section VI. Note that this classifier is not necessarily optimal for a fixed topology. We conducted a number of experiments of 10,000

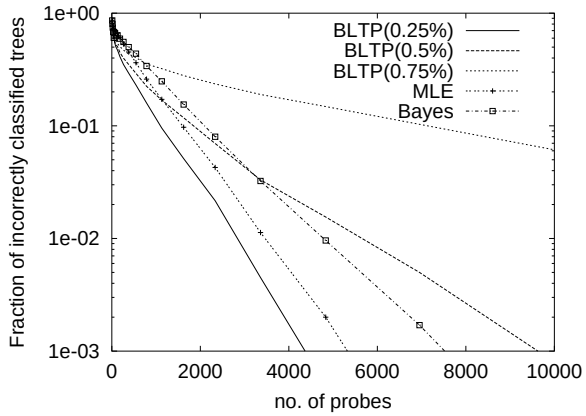


(a) random topology



(b) random topology

Fig. 16. MISCLASSIFICATION IN ML, BAYESIAN AND BLT CLASSIFIER: (τ, α) RANDOMLY DRAWN ACCORDING TO THE PRIOR DISTRIBUTION. (a) Bayes and BLTP(ε) classifier. (b) Bayes and ML classifiers.



fixed topology

	Expt.	Approx.
BLTP(0.25%)	0.00158	0.00130
BLTP(0.5%)	0.00048	0.00058
BLTP(0.75%)	0.00014	0.00014
ML	0.00105	0.00079

Fig. 17. MISCLASSIFICATION IN ML, BAYESIAN AND BLT CLASSIFIER: FIXED (τ, α). LEFT: fraction of misclassified topologies. RIGHT: Comparison of experimental and approximated tail slopes.

simulations of the three algorithms with fixed loss trees. The relative accuracy of the algorithms was found to vary with both topology and link loss rates. However, in all example we found a value of ε for which BLTP(ε) accuracy either closely approached or exceeded that of the ML and Bayesian classifiers. As an example, in Figure 17 we plot the results for the first binary tree topology in Figure 15 with all loss rates equal to 10% but that of the sole internal link, which has loss rate 1%. In this example, the ML classifier is more accurate than the pseudo-Bayesian classifier. BLTP(ε) accuracy improves as ε is decreased, and eventually, for $\varepsilon = 0.25\%$, it exceeds that of the pseudo-Bayesian and ML classifier.

These experimental results are supported by approximations to the tail slopes of the log misclassification probabilities, as detailed in Section VIII. For the same example, we display in Figure 17 (right), the estimated experimental and numerical approximated tail slopes of the ML and BLTP classifiers. For a given classifier these agree within about 25%. Finally, not reported in the Figure, we also verified that the ML(ε) classifiers provide the same accuracy as BLTP(ε).

D. Summary.

Whereas the Bayesian classifier is optimal in the context of a random topology with known prior distribution, similar accuracy can be achieved using BLTP(ε) or ML(ε) with an appropriately chosen threshold ε. In fixed topologies, the corresponding pseudo-Bayes classifier is not necessarily optimal. In the fixed topologies for which we were able to make comparisons, better accuracy could be obtained using BLTP(ε) or ML(ε) with an appropriate threshold ε. The accuracy of BLTP(ε) and ML(ε) are similar: most of the time BLTP selects the ML topology with maximum likelihood.

BLTP has the lowest complexity, primarily because

each grouping operation excludes subsets of candidate topologies from further consideration. By contrast, the ML and Bayesian classifiers used exhaustive searches through the space of possible topologies. Since the number of possible topologies grows rapidly with the number of receivers, these methods have high complexity. A more sophisticated search strategy could reduce complexity for these classifiers, but we expect this to be effective only if the number of topologies to be searched is reduced (e.g. in the manner of BLTP). With larger numbers of receivers, any fixed reduction in the per-topology computational complexity would eventually be swamped due to the growth in the number of possible topologies.

VIII. Misgrouping and Misclassification

In this section, we analyze more closely the modes of failure of BLTP, and estimate convergence rates of the probability of correct classification. Since this classifier proceeds by recursively grouping receivers, we can analyze topology misclassification by looking at how sets of receivers can be misgrouped in the estimated topology $\widehat{\mathcal{T}}$. We formalize the notion of correct receiver grouping as follows. $R_{\mathcal{T}}$ will denote the set of receivers in the logical multicast topology $\mathcal{T}$.

Definition 1: Let $(\mathcal{T}, \alpha)$ be a loss tree with $\mathcal{T} = (V, L)$, and let $(\widehat{\mathcal{T}}, \widehat{\alpha})$ be an inferred loss tree with $\widehat{\mathcal{T}} = (\widehat{V}, \widehat{L})$. The receivers $R_{\mathcal{T}}(i)$ descended from a node $i \in W$ are said to be correctly grouped in $\widehat{\mathcal{T}}$ if there exists a node $\widehat{\imath} \in \widehat{V}$ such that $R_{\mathcal{T}}(i) = R_{\widehat{\mathcal{T}}}(\widehat{\imath})$. In this case we shall say also that node i is correctly classified in $\widehat{\mathcal{T}}$.

Observe that we allow the trees rooted at i and $\widehat{\imath}$ to be different in the above definition; we only require the two sets of receivers to be equal.

Correct receiver grouping and correct topology classification are related: in the case of binary trees, the topology is correctly classified if and only if every node $k \in W$ is correctly classified. This allows us to study topology misclassification by looking at receiver misgrouping. To this end, we need to first introduce a more general form of the function $B(\cdot)$ to take into account expressions which may arise as result of classification errors. Observe that in (6) for $k \in V$ we defined $Y_k^{(i)}$ as $Y_k^{(i)} = \vee_{j \in d(k)} Y_j^{(i)} = \vee_{j \in R_{\mathcal{T}}(k)} Y_j^{(i)}$. In line 9 of BLT we have for the newly formed node U, $Y_U^{(i)} = \vee_{u \in U} Y_u^{(i)} = \vee_{j \in S} Y_j^{(i)}$, for some subset S of $R_{\mathcal{T}}$. By construction S is the set of receivers of the subtree of $\widehat{\mathcal{T}}$ rooted in U (which has been obtained by recursively grouping the nodes in S). It is clear that $S = R_{\mathcal{T}}(k)$ for some node $k \in V$ if the subtree has been correctly reconstructed, but, upon an error, can be otherwise a generic subset of $R_{\mathcal{T}}$. Therefore, in BLT we need to consider the following more general expression

$$\begin{aligned} \widehat{B}(S_1, S_2) &:= \frac{\sum_{i=1}^n \vee_{j \in S_1} Y_j^{(i)} \sum_{i=1}^n \vee_{j \in S_2} Y_j^{(i)}}{n \sum_{i=1}^n (\vee_{j \in S_1} Y_j^{(i)}) \cdot (\vee_{j \in S_2} Y_j^{(i)})} \\ &= \frac{\widehat{\gamma}(S_1)\widehat{\gamma}(S_2)}{\widehat{\gamma}(S_1) + \widehat{\gamma}(S_2) - \widehat{\gamma}(S_1 \cup S_2)} \quad (14) \end{aligned}$$

where S_1 and S_2 are two non empty disjoint subsets of $R_{\mathcal{T}}$. Analogous to Theorem 4, $\lim_{n \to \infty} \widehat{B}(S_1, S_2) = B(S_1, S_2)$, where

$$\begin{aligned} B(S_1, S_2) &:= \frac{\mathsf{P}[\vee_{j \in S_1} X_j = 1]\mathsf{P}[\vee_{j \in S_2} X_j = 1]}{\mathsf{P}[\vee_{j \in S_1} X_j \cdot \vee_{j \in S_2} X_j = 1]} \\ &= \frac{\gamma(S_1)\gamma(S_2)}{\gamma(S_1) + \gamma(S_2) - \gamma(S_1 \cup S_2)}. \quad (15) \end{aligned}$$

(15) can be regarded as a generalization of (5) where we consider a pair of disjoint sets of receivers instead of pair of nodes.

A. Misgrouping and Misclassification in BLT

We start by studying misgrouping in binary trees under BLT. Consider the event G_i that BLT correctly groups nodes in $R_{\mathcal{T}}(i)$ for some $i \in W$. This happens if grouping operations do not pair any nodes formed by recursive grouping $R_{\mathcal{T}}(i)$, with any nodes formed similarly from the complement $R_{\mathcal{T}} \setminus R_{\mathcal{T}}(i)$, until no candidate pairs in $R_{\mathcal{T}}(i)$ remain to be grouped.

Lemma 1: A sufficient condition for correct grouping of i is that

$$\widehat{D}(S_1, S_2, S_3) := \widehat{B}(S_1, S_3) - \widehat{B}(S_1, S_2) > 0 \quad (16)$$

for all $(S_1, S_2, S_3) \in \mathcal{S}(i) = \{(S_1, S_2, S_3) : S_1, S_2 \subset R_{\mathcal{T}}(i), S_3 \subseteq R_{\mathcal{T}} \setminus R_{\mathcal{T}}(i), S_k \neq \emptyset, k = 1, 2, 3, S_k \neq S_\ell, k \neq \ell\}$.

Therefore $G_i \supseteq Q_i = \cap_{(S_1, S_2, S_3) \in \mathcal{S}(i)} Q(S_1, S_2, S_3)$ where $Q(S_1, S_2, S_3)$ denotes the event that (16) holds. This provides the following upper bound for probability of misgrouping i, denoted by

$$P_i^f := \mathsf{P}[G_i^c] \leq \sum_{(S_1, S_2, S_3) \in \mathcal{S}(i)} \mathsf{P}[Q^c(S_1, S_2, S_3)] \quad (17)$$

A.1 Estimation of Misclassification Probabilities.

We now consider the asymptotic behavior of P_i^f, first for large n, then for small loss probabilities $\overline{\alpha} = 1 - \alpha$. Let $s(k) := \sum_{l \succeq k} \overline{\alpha}_l$, $k \in V$, and set $D(\cdot) = \mathsf{E}[\widehat{D}(\cdot)]$.

Theorem 10: Let $(\mathcal{T}, \alpha)$ be a canonical loss tree. For each $i \in W$, $\sqrt{n} \cdot (\widehat{D}(S_1, S_2, S_3) - D(S_1, S_2, S_3))$, $(S_1, S_2, S_3) \in \mathcal{S}(i)$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2(S_1, S_2, S_3)$, with

$D(S_1, S_2, S_3) = B(S_1, S_3) - B(S_1, S_2)$. Moreover, as $\|\overline{\alpha}\| = \max_{k \in V} \overline{\alpha}_k \to 0$, then:
(i) $D(S_1, S_2, S_3) = s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3)) + O(\|\overline{\alpha}\|^2)$;
(ii) $\sigma^2(S_1, S_2, S_3) = s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3)) + O(\|\overline{\alpha}\|^2)$;
(iii)

$$\min_{(S_1,S_2,S_3) \in \mathcal{S}(i)} \frac{D^2(S_1, S_2, S_3)}{\sigma^2(S_1, S_2, S_3)} = \overline{\alpha}_i + O(\|\overline{\alpha}\|^2), \quad (18)$$

where, for small enough $\|\overline{\alpha}\|$, the minimum is attained for S_1, S_2, S_3 such that $a(S_1 \cup S_2) = i$ and $a(S_1 \cup S_3) = f(i)$.

Theorem 10 suggests we approximate $\mathsf{P}[Q^c(S_1, S_2, S_3)]$ by $\Psi\left(-\sqrt{n} \cdot \frac{D(S_1,S_2,S_3)}{\sigma(S_1,S_2,S_3)}\right)$, where Ψ is the cdf of the standard normal distribution. Thus for large n and small $\|\overline{\alpha}\|$, Theorem 10 and (17) together suggest that we approximate the misgrouping probability

$$P_i^f \approx e^{-\overline{\alpha}_i \frac{n}{2}} \quad (19)$$

Here we have used the fact that the P_i^f should be dominated by the summand with the smallest (negative) exponent according to (18). Thus, asymptotically for many probes, the probability of correctly identifying a group of receivers descended from node i is determined by the loss rate of link i alone, and is larger for lossier links. Moreover, the stated relations between the minimizing (S_1, S_2, S_3) in (iii) say that the likely mode of failure is to mistakenly group a child of i with the sibling of i.

In binary trees, the topology is correctly classified when all groups are correctly formed. Hence $P^f_{\mathsf{BLT}} \leq \sum_{i \in W} P_i^f \approx \max_{i \in W} P_i^f$, and we expect $\log P^f_{\mathsf{BLT}}$ to be an asymptotically linear with function of n with negative slope $\overline{\alpha}^f/2$, where

$$\overline{\alpha}^f = \min_{i \in W} \overline{\alpha}_i. \quad (20)$$

Thus, in the regime considered, the most likely way to misclassify a tree is by incorrectly grouping siblings whose parent node j terminates the least lossy internal link, mistakenly grouping the sibling of j with one of its children.

We remark that the preceding argument can be formalized using Large Deviation theory [5]. However, calculation of the decay rate appears computationally infeasible, although one can recover the leading exponent $\overline{\alpha}^f/2$ in the small $\|\overline{\alpha}\|$ regime.

A.2 Experimental Evaluation.

Although we have derived the slope $\overline{\alpha}^f$ through a series of approximations, we find that it describes experimental misclassification and misgrouping reasonably well. We performed 10,000 experiments with an eight-leaf perfectly balanced binary tree. On each experiment, the loss rates are a random permutation of the elements of the set $\{0.5\%, 1\%, \ldots, 7\%, 7.5\%\}$. In this way, the smallest loss rate is fixed to 0.5%. In Figure 18 we plot the proportion of links, that had loss rates greater than or equal to a given threshold ϕ, and were misclassified. As the number of probes increases, misclassification is due exclusively to misgrouping of low loss rate links: in this set of experiments, no link with loss rate higher than 2% was misclassified once the number of probes exceeded 700.

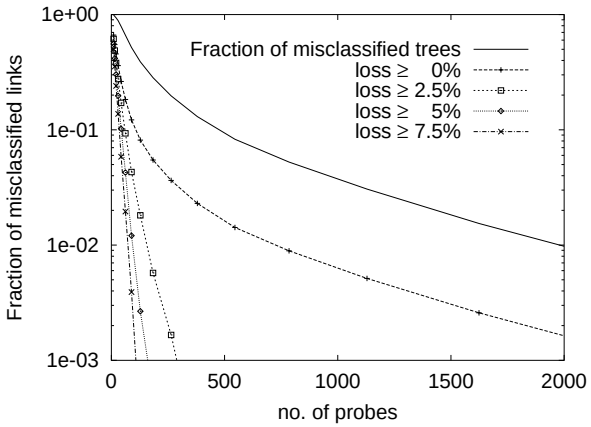
According to (19), the different curves should be asymptotically linear with negative slope approximately $\phi/2$ (then adjusted by a factor $\log_{10} e$ since the logarithms are to base 10). On the table in Figure 18(right) we display the estimated experimental and approximated slopes. Agreement is good for $\phi = 2.5\%$ and 5%. We believe the greater error for $\phi = 7.5\%$ may be due to the departure from the leading order linear approximations of (18) for larger values of $\overline{\alpha}_k$; also relatively few points are available for estimation from the experimental curves. In the figure, we also plot the log fraction of times BLT correctly identify the topology; as expected, this curve exhibits the same asymptotic linear slope of the fraction of misgrouped links, i.e., the one for $\phi = 0\%$.

B. Misgrouping and Misclassification in BLTP(ε)

We turn our attention to the errors in classifying general trees by the reference algorithm BLTP(ε). In the following, without loss of generality, we will study the errors in the classification of the pruned tree $(\mathcal{T}^\varepsilon, \alpha^\varepsilon) = \mathsf{TP}(\varepsilon)(\mathcal{T}, \alpha)$, with $\mathcal{T}^\varepsilon = (V^\varepsilon, L^\varepsilon)$, under the assumption that $\varepsilon \neq \alpha_k$, $k \in W$. This will include, as a special case, when ε is smaller than the internal link loss rates of the underlying tree, i.e., when $\mathcal{T}^\varepsilon = \mathcal{T}$, the analysis of the misclassification of $\mathcal{T}$. $W^\varepsilon = V^\varepsilon \setminus (\{0,1\} \cup R_{\mathcal{T}^\varepsilon})$ will denote the set of nodes in $\mathcal{T}^\varepsilon$ terminating internal links.

Let $(\widehat{\mathcal{T}}, \widehat{\alpha})$ denote the tree produced by BLT, the final estimate $\widehat{\mathcal{T}}^\varepsilon$ is obtained from $\widehat{\mathcal{T}}$ by pruning links whose loss rate is smaller or equal than ε, i.e., $(\widehat{\mathcal{T}}^\varepsilon, \widehat{\alpha}^\varepsilon) = \mathsf{TP}(\varepsilon)(\widehat{\mathcal{T}}, \widehat{\alpha})$. In contrast to the binary case, incorrect grouping by BLT may be sufficient but not necessary for misclassification. For BLTP(ε), incorrect classification occurs if any of the following hold:
(i) at least one node in $\mathcal{T}^\varepsilon$ is misclassified in $\widehat{\mathcal{T}}^\varepsilon$;
(ii) $\mathsf{TP}(\varepsilon)$ prunes links from $\widehat{\mathcal{T}}$ that are present in $\mathcal{T}^\varepsilon$; or
(iii) $\mathsf{TP}(\varepsilon)$ fails to prune links from $\widehat{\mathcal{T}}$ that are not present in $\mathcal{T}^\varepsilon$.
Observe that (i) implies that a node i such that $\overline{\alpha}_i \leq \varepsilon$ can be misclassified and still $\widehat{\mathcal{T}}^\varepsilon = \mathcal{T}^\varepsilon$ provided the all the resulting erroneous links are pruned.

We have approximated the probability of errors of type (i) in our analysis of BLT. Errors of type (ii) are excluded



ϕ	Expt.	Approx.
0%	0.0005	0.0011
2.5%	0.0051	0.0054
5.0%	0.0097	0.0109
7.5%	0.0248	0.0163

Fig. 18. MISCLASSIFICATION AND MISGROUPING IN BLT. LEFT: fraction of links misclassified with loss $\geq \phi$, for $\phi = 0\%, 2.5\%, 5.0\%.7.5\%$. RIGHT: Comparison of experimental and approximated tail slopes.

if for all $i \in W^\varepsilon$:

$$\widehat{D}(S_1,S_2,S_3,\varepsilon) := \widehat{B}(S_1 \cup S_2, S_3)(1-\varepsilon) - \widehat{B}(S_1,S_2) > 0 \quad (21)$$

for all $(S_1,S_2,S_3) \in \mathcal{S}(i)$, since this condition implies that all estimated loss rates of links in the actual tree are greater than ε. Errors of type (iii) are excluded if $\widehat{B}(S_1,S_3) - \widehat{B}(S_1,S_2) > 0$ and $\widehat{B}(S_1 \cup S_2, S_3)(1-\varepsilon) - \widehat{B}(S_1,S_2) \leq 0$, or if $\widehat{B}(S_1,S_3) - \widehat{B}(S_1,S_2) \leq 0$ and $\widehat{B}(S_1 \cup S_3, S_2)(1-\varepsilon) - \widehat{B}(S_1,S_3) \geq 0$ for all $(S_1,S_2,S_3) \in \mathcal{S}(\varepsilon)$ where $\mathcal{S}(\varepsilon) = \{(S_1,S_2,S_3) : S_j \subset R;\ S_j \neq \emptyset;\ S_j \cap S_k = \emptyset, j \neq k;\ (S_1 \cup S_2 \cup S_3) \cap R_{\mathcal{T}}(i) = \emptyset \vee (S_1 \cup S_2 \cup S_3) \subseteq R_{\mathcal{T}}(i) \vee \exists j \in \{1,2,3\}\, R_{\mathcal{T}}(i) \subseteq S_j, i \in W^\varepsilon\}$. The latter conditions ensure that all the links in the binary tree produced by BLT, which are either results of node misgrouping or corresponding to fictitious links due to binary reconstruction, have estimated loss rate less than or equal to ε, and are hence pruned. Summarizing, let $Q(S_1,S_2,S_3,\varepsilon)$ be the event that (21) holds for a given (S_1,S_2,S_3), and $G(\varepsilon)$ the event that the topology is correctly classified. Then $G(\varepsilon) \supseteq \cap_{k \in V^\varepsilon \setminus R_{\mathcal{T}}}(Q_k \cap Q_k^{(ii)}(\varepsilon)) \cap Q^{(iii)}(\varepsilon)$ where $Q_k^{(ii)}(\varepsilon) = \cap_{(S_1,S_2,S_3)\in\mathcal{S}(k)} Q(S_1,S_2,S_3,\varepsilon)$ and $Q^{(iii)}(\varepsilon) = \cap_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)}(Q(S_1,S_2,S_3) \cap Q(S_1,S_2,S_3,\varepsilon)^c) \cup (Q(S_1,S_2,S_3)^c \cap Q(S_1,S_3,S_2,\varepsilon)^c)$. Consequently, we can write a union bound for the probability of misclassification:

$$P^f_{\mathsf{BLTP}(\varepsilon)} := P[G(\varepsilon)^c] \quad (22)$$
$$\leq \sum_{k \in W^\varepsilon} \left(\mathsf{P}[Q_k^c] + \mathsf{P}[Q_k^{(ii)}(\varepsilon)^c] \right) + \mathsf{P}[Q^{(iii)}(\varepsilon)^c]$$

and each term in (22) can in turn be bounded above by a sum similar to the RHS of (17). For the last term, in particular, observe that

$$Q^{(iii)}(\varepsilon)^c = \cup_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)}(Q(S_1,S_2,S_3)^c \cup \quad (23)$$
$$Q(S_1,S_2,S_3,\varepsilon)) \cap (Q(S_1,S_2,S_3) \cap Q(S_1,S_3,S_2,\varepsilon))$$
$$\subseteq \cup_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)}(Q(S_1,S_2,S_3,\varepsilon) \cup Q(S_1,S_3,S_2,\varepsilon))$$
$$= \cup_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)} Q(S_1,S_2,S_3,\varepsilon),$$

so that $\mathsf{P}[Q^{(iii)}(\varepsilon)^c] \leq \sum_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)} Q(S_1,S_2,S_3,\varepsilon)$.

B.1 Misclassification Probabilities and Experiment Duration.

We examine the asymptotics of the misclassification probability $P^f_{\mathsf{BLTP}(\varepsilon)}$ for large n and small $\|\overline{\alpha}\|$, by the same means as in Section VIII-A. This amounts to finding the mean $D(S_1,S_2,S_3,\varepsilon)$ and asymptotic variance $\sigma^2(S_1,S_2,S_3,\varepsilon)$ of the distribution of $\widehat{D}(S_1,S_2,S_3,\varepsilon)$, then finding the dominant exponent D^2/σ^2 over the various (S_1,S_2,S_3). Let $\overline{\alpha}^f(\varepsilon) = \min_{i\in W^\varepsilon} \overline{\alpha}_i$ denote the smallest internal link loss rate of $\mathcal{T}^\varepsilon$ larger than ε and $\overline{\alpha}^p(\varepsilon) = \max_{i \in W \setminus W^\varepsilon} \overline{\alpha}_i$ the largest internal link loss rate of $\mathcal{T}$ smaller than ε or $\overline{\alpha}^p(\varepsilon) = 0$ if no such loss rate exists (which occurs when ε is smaller than all internal links loss rate). The proof of the following result is similar to that of Theorem 10 and is omitted.

Theorem 11: Let $(\mathcal{T},\alpha)$ be a canonical loss tree. For each $0 \leq \varepsilon < 1$, $(S_1,S_2,S_3) \in \cup_{i\in W^\varepsilon}\mathcal{S}(i) \cup \mathcal{S}(\varepsilon)$, $\sqrt{n} \cdot (\widehat{D}(S_1,S_2,S_3,\varepsilon) - D(S_1,S_2,S_3,\varepsilon))$ converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2(S_1,S_2,S_3,\varepsilon)$. Furthermore, as $\|\overline{\alpha}\| = \max_{k\in V} \overline{\alpha}_k \to 0$ and $\varepsilon/\|\overline{\alpha}\| \to c \in (0,\infty)$,

(i) $D(S_1,S_2,S_3,\varepsilon) = s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3)) - \varepsilon + O(\|\overline{\alpha}\|^2)$;

(ii) $\sigma^2(S_1,S_2,S_3,\varepsilon) = |s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3))| + O(\|\overline{\alpha}\|^2)$;

(iii) If $(S_1, S_2, S_3) \in \mathcal{S}(i)$, $i \in W^\varepsilon$,

$$\min_{(S_1,S_2,S_3)\in\mathcal{S}(i)} \frac{D^2(S_1,S_2,S_3,\varepsilon)}{\sigma^2(S_1,S_2,S_3,\varepsilon)} = \frac{(\overline{\alpha}_i-\varepsilon)^2}{\overline{\alpha}_i} + O(\|\overline{\alpha}\|^2) \tag{24}$$

and

$$\min_{i\in W^\varepsilon} \min_{(S_1,S_2,S_3)\in\mathcal{S}(i)} \frac{D^2(S_1,S_2,S_3,\varepsilon)}{\sigma^2(S_1,S_2,S_3,\varepsilon)} = \frac{(\overline{\alpha}^f(\varepsilon)-\varepsilon)^2}{\overline{\alpha}^f(\varepsilon)} + O(\|\overline{\alpha}\|^2). \tag{25}$$

If $(S_1, S_2, S_3) \in \mathcal{S}(\varepsilon)$,

$$\min_{(S_1,S_2,S_3)\in\mathcal{S}(\varepsilon)} \frac{D^2(S_1,S_2,S_3,\varepsilon)}{\sigma^2(S_1,S_2,S_3,\varepsilon)} = \begin{cases} O(\varepsilon^2/\|\overline{\alpha}\|^2) & \text{if } \overline{\alpha}^p(\varepsilon)=0 \\ \frac{(\varepsilon-\overline{\alpha}^p(\varepsilon))^2}{\overline{\alpha}^p(\varepsilon)} + O(\|\overline{\alpha}\|^2) & \text{if } \overline{\alpha}^p(\varepsilon)>0 \end{cases} \tag{26}$$

In (27) above, for clarity we distinguish the expressions for $\overline{\alpha}^p(\varepsilon) = 0$ and $\overline{\alpha}^p(\varepsilon) > 0$. Observe that the result for $\overline{\alpha}^p(\varepsilon) = 0$ in (27) can be actually obtained by taking the limit of the expression for $\overline{\alpha}^p(\varepsilon) > 0$, which is of the form $\Big((\varepsilon-\overline{\alpha}^p(\varepsilon))^2 + O(\|\overline{\alpha}\|^3)\Big) \,/\, \Big(\overline{\alpha}^p(\varepsilon) + O(\|\overline{\alpha}\|^2)\Big)$.

Using the same reasoning as was used in Section VIII-A, we expect that the logarithms of the probabilities of errors of type (i), (ii) and (iii) to be asymptotically linear in the number of probes n, with slopes that behave respectively as

$$c^{(i)} = \overline{\alpha}^f(\varepsilon)/2, \quad c^{(ii)} = \frac{(\overline{\alpha}^f(\varepsilon)-\varepsilon)^2}{2\overline{\alpha}^f(\varepsilon)}, \tag{27}$$

$$c^{(iii)} = \begin{cases} O(\varepsilon^2/\|\overline{\alpha}\|^2) & \text{if } \overline{\alpha}^p(\varepsilon)=0 \\ \frac{(\varepsilon-\overline{\alpha}^p(\varepsilon))^2}{2\overline{\alpha}^p(\varepsilon)} & \text{if } \overline{\alpha}^p(\varepsilon)>0 \end{cases}$$

The dominant mode of misclassification is that with the lowest slope in (27), which then dominates the sum in (22) for large n. Hence we approximate the misclassification probability to leading exponential order by

$$P^f_{\mathsf{BLTP}(\varepsilon)} \approx e^{-(n/2)\min\{c^{(i)},c^{(ii)},c^{(iii)}\}}. \tag{28}$$

Since $c^{(i)} \geq c^{(ii)}$, type (ii) errors always dominate type (i). Between type (ii) and (iii), the prevailing type of errors depends on the relative magnitude of $\overline{\alpha}^f(\varepsilon)$, $\overline{\alpha}^p(\varepsilon)$ and ε, which satisfy $\overline{\alpha}^p(\varepsilon) < \varepsilon < \overline{\alpha}^f(\varepsilon)$. Type (ii) becomes prevalent as $\varepsilon \to \overline{\alpha}^f(\varepsilon)$ since then $c^{(ii)} \to 0$; similarly, type (iii) dominates as $\varepsilon \to \overline{\alpha}^p(\varepsilon)$. Thus, ε should be chosen large enough to avoid the type (iii) errors, but small enough so that the probability of type (ii) does not become large. Unfortunately, this is not possible unless information on the actual link loss rates is available. We believe, nevertheless, that this does not represent a problem in practice. Indeed, as the analysis above indicates,

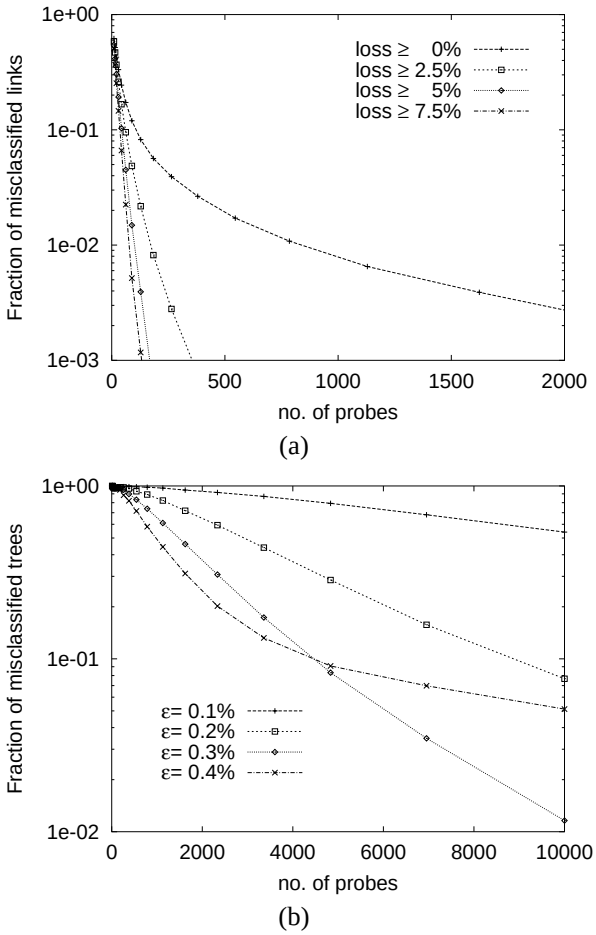


Fig. 19. MISCLASSIFICATION AND MISGROUPING IN $\mathsf{BLTP}(\varepsilon)$: (a) fraction of misclassified links with loss $\geq \phi$, for $\phi = 0\%, 2.5\%, 5.0\%.7.5\%$; (b) fraction of misclassified trees for $\varepsilon = 0.1\%, 0.2\%, 0.3\%, 0.4\%$.

for enough large n, the most likely way $\mathsf{BLTP}(\varepsilon)$ misclassify a tree is by either pruning the link which the least loss rate higher than ε (a type (ii) error) or by not pruning that with the the largest loss rate smaller than ε (a type (iii) error); either way, the resulting inferred tree would differ from the actual by the at most one link, approximatively, that with the loss rate closest to ε.

The foregoing arguments allow us to also estimate the number of probes N required for inference with misclassification probability δ in a tree with minimum link loss rate $\overline{\alpha}^f$. This is done by inverting the approximation (28) to obtain that N is approximately

$$\begin{cases} -\frac{2\overline{\alpha}^f(\varepsilon)\log\delta}{(\varepsilon-\overline{\alpha}^f(\varepsilon))^2} & \text{if } \overline{\alpha}^p(\varepsilon)=0 \\ -2\max\left\{\frac{\overline{\alpha}^f(\varepsilon)}{(\varepsilon-\overline{\alpha}^f(\varepsilon))^2}, \frac{\overline{\alpha}^p(\varepsilon)}{(\varepsilon-\overline{\alpha}^p(\varepsilon))^2}\right\}\log\delta & \text{if } \overline{\alpha}^p(\varepsilon)>0 \end{cases} \tag{29}$$

Note that for BLT, or when $\varepsilon \ll \overline{\alpha}^f$, this reduces to the simple form $N \approx -2\log(\delta)/\overline{\alpha}^f$.

We conclude by observing that in the above analysis, we have implicitly assumed that $W^\varepsilon \neq \emptyset$. Nevertheless, for large enough ε, $W^\varepsilon = \emptyset$ which corresponds to the case when $\mathcal{T}^\varepsilon$ is a degenerate tree where all leaf nodes are siblings. In this case, it is clear that misclassification occurs only because of type (iii) errors. The misclassification analysis for this special case can then be obtained by taking into account type (iii) errors alone.

B.2 Experimental Evaluation.

We performed 10,000 experiments in a 21 node tree with mixed branching ratio 2 and 3. On each experiment, the loss rates are a random permutation of the elements of the set $\{0.5\%, 1\%, \ldots, 9.5\%, 10\%\}$, thus having the same smallest link loss as in the experiments for BLT. In Figure 19 we plot the fraction of links, that had loss rates greater than or equal to a given threshold ϕ, and were misclassified. These appear very similar to those for BLT in Figure 18. In Figure 19(b) we also plot the fraction of misclassified trees using $\mathsf{BLTP}(\varepsilon)$ for different values of ε, all smaller than the smallest loss rate of 0.5%. With this choice, $\overline{\alpha}^p(\varepsilon) = 0$ and $\overline{\alpha}^f(\varepsilon) = 0.5\%$. As expected, accuracy is best for intermediate ε. The difference in shape between the last and the first three curves indicates the change between the two different regimes of misclassification. For ε smaller than 0.4%, misclassification is dominated by erroneous exclusion of nodes from a group, while for $\varepsilon = 0.4\%$, misclassification is mostly determined by erroneous pruning of the link with the smallest loss rate (which is 0.5%) because of statistical fluctuation of its inferred loss rate below ε. In the latter case, we can use (27) to compute the tail slope obtaining 4.3×10^{-4}, in good agreement with the estimated experimental slope which is 4.1×10^{-4}.

B.3 Asymptotic Misclassification Rates for the ML-Classifier

We sketch how the theory of large deviations [5] can be used to bound the asymptotic probability of misclassification by the ML estimator. The expressions obtained here were used to determine the ML tail slopes in the table in Figure 17. First, observe that $\mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\mathrm{ML}} \neq \mathcal{T}) = \sum_{\tau \neq \mathcal{T}} \mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\mathrm{ML}} = \tau)$. For $\tau \neq \mathcal{T}$, each term in this sum can be bounded above by $\mathsf{P}_{\mathcal{T},\alpha}\left(\cup_{\alpha' \in \mathcal{A}_\tau} \{n^{-1} \sum_{i=1}^n g(X^{(i)}; \tau, \alpha') > 0\}\right)$, where $g(x; \tau, \alpha') = \log(p(x; \tau, \alpha')/p(x; \mathcal{T}, \alpha))$ and $p(x; \mathcal{T}, \alpha)$ the probability of the outcome $x \in \Omega = \{0,1\}^R$ under the loss tree $(\mathcal{T}, \alpha)$. Let $\mu_n = n^{-1} \sum_{i=1}^n \delta_{X^{(i)}}$ denote the empirical distribution of the first n quantities $X^{(i)}$ (here δ_x is the unit mass at x), and for each τ and $\alpha' \in \mathcal{A}_\tau$ let $\Gamma_{\tau,\alpha'} = \{\nu \in M_1(\Omega) : \sum_{x \in \Omega} g(x; \tau, \alpha')\nu(x) \geq 0\}$ (here $M_1(\Omega)$ is the set of probability measures on Ω) and set $\Gamma_\tau = \cup_{\alpha' \in \mathcal{A}_\tau} \Gamma_{\tau,\alpha'}$. Since the $g(X^{(i)}, \tau, \alpha')$ are IID random variables, we can use Sanov's Theorem [5] to conclude that

$$\begin{aligned} \limsup_{n\to\infty} \frac{1}{n} \log \mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\mathrm{ML}} = \tau) \\ \leq \limsup_{n\to\infty} \frac{1}{n} \log \mathsf{P}_{\mathcal{T},\alpha}(\mu_n \in \Gamma_\tau) \\ \leq - \inf_{\nu \in \Gamma_\tau} K(\nu \mid f(\cdot; \mathcal{T}, \alpha)). \end{aligned} \tag{30}$$

Here, for $\nu, \eta \in M_1(\Omega)$, $K(\nu \mid \eta) = \sum_{x \in \Omega} \nu(x) \log(\nu(x)/\eta(x))$ is the Kullback-Leibler "distance", or entropy of ν relative to η. By further minimizing the right-hand term of (30) over all $\tau \neq \mathcal{T}$, we obtain an asymptotic upper bound for the decay rate of the misclassification probability as n increases. For each τ, the minimization can be carried out using the Kuhn-Tucker theorem; we use the form given in [15].

We mention that a lower bound of the following form can be found:

$$\begin{aligned} \liminf_{n\to\infty} \frac{1}{n} \log \mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\mathrm{ML}} \neq \mathcal{T}) \quad \geq \\ - \inf\{K(\tau', \alpha' | \mathcal{T}, \alpha) : \tau' \neq \mathcal{T}, \alpha' \in \mathcal{A}_{\tau'}\} \end{aligned} \tag{31}$$

IX. Summary and Conclusions

In this paper we have proposed and established the consistency of a number of algorithms for inferring logical multicast topology from end-to-end multicast loss measurements. The algorithms fall in two broad classes: the grouping algorithms (BLTP, BLTC and GLT), and the global algorithms (ML and Bayesian).

The computational cost of the grouping approaches is considerably less for two reasons: (i) they work by progressively excluding subsets of candidate topologies from consideration while the global algorithms inspect all topologies; and (ii) their cost per inspection of each potential sibling set is lower. Of the grouping algorithms, the BLTP approach of treating the tree as binary then pruning low loss links is simplest to implement and execute.

Of the algorithms presented, only the Bayesian is able to identify links with arbitrarily small loss rates. All the other classifiers require a parameter $\varepsilon > 0$ that acts as a threshold: a link with loss rate below this value will be ignored and its endpoints identified. The threshold is required in order that sibling groups not be separated due to random fluctuations of the inferred loss rates. However, we do not believe that the necessity of a threshold presents an obstacle to their use in practice, since it is the identification of high loss links that is more important for performance diagnostics. In practice we expect ε to be chosen according to an application-specific notion of a minimum relevant loss rate.

By construction, the Bayesian classifier has the greatest accuracy in the context of classification of topologies

drawn according to a known random distribution. However, the performance gap narrows when classifying a fixed unknown topology, and in fact the Bayesian classifier has slightly worse performance than the others in this context. We conclude that BLTP offers the best performance, having the lowest computational cost for near optimal performance.

This selection of BLTP(ε) motivates analyzing its error modes, and their probabilities. Although the analysis is quite complex, a simple picture emerges in the regime of small loss rates $\overline{\alpha}_k$ and many probes n, and errors are most likely to occur when grouping the children of the node j that terminates the link of lowest loss rate.

The leading exponents for the misclassification that were calculated in Section VIII can be used to derive rough estimates of the number of probes required in practice. Consider the problem of classifying a general topology whose smallest link loss rate in 1%. According to (29), the number of probes required for a misclassification probability of 1% (using $\varepsilon = 0.5\%$) is about 4000. (In a binary topology using BLT the number required drops to about 1000). Using small (40 byte) probes at low rate of a few tens of kbits per sec, measurements involving this many probes could be completed within only a few minutes.

We note that the grouping methods extend to a wider class of estimators by replacing the shared loss estimate with any function on the nodes (i) that increases on moving away from the root; and (ii) whose value at a node can be consistently estimated from measurements at receivers descended from that node. Examples of such quantities include the mean and variance of the cumulative delay from the root to a given node; see [6] and [11].

Finally, a challenging problem is to take the resulting logical multicast trees and mapping the constituent nodes onto physical routers within real networks. This remains beyond our capability at this time.

X. Proofs of the Theorems

The proof of Proposition 1 depends in the following Lemma.

Lemma 2: Let $g_i > 0$ for $i = 1, 2, \ldots, m$; let g be such that $\min_i g_i < g < \sum_i g_i$; and set $f(b) := (1 - g/b) - \prod_i (1 - g_i/b)$. Then the equation $f(b) = 0$ has a unique solution $b^* > g$. Furthermore, given $b > g$ then $f(b) > 0$ if and only if $b > b^*$.

Proof of Lemma 2: Set $c_i = g_i/g < 1$ so that $\sum_i c_i > 1$. Let $h_1(x) = (1 - x)$, $h_2(x) = \prod_i (1 - c_i x)$ and $h = h_1 - h_2$, so that $f(b) = h(g/b)$. We look for zeroes of h. For $x \in [0, 1]$ $h_1''(x) = 0$, $h_2''(x) = h_2(x) \left\{ \left(\sum_i q_i(x) \right)^2 - \sum_i q_i(x)^2 \right\} > 0$ where $q_i(x) = c_i/(1 - c_i x) > 0$. Hence h is strictly concave on $[0, 1]$. Now $h(0) = 0$, $h(1) < 0$ and $h'(0) = -1 + \sum_i c_i > 0$. So since h is concave and continuous on $[0, 1]$ there must be exactly one solution x^* to $h(x) = 0$ for $x \in (0, 1)$ and hence one solution b^* to $f(b) = 0$ for $b > g$. Furthermore, given $x \in (0, 1)$, $h(x) > 0$ iff $x < x^*$ and hence given $b > g$, $f(b) > 0$ iff $b > b^*$. ■

Proof of Proposition 1: Clearly $\min_{k \in U} \gamma(k) < \gamma(U) < \sum_{k \in U} \gamma(U)$ in a canonical loss tree and hence (i) and (ii) follow from Lemma 2. (iii) is then a restatement of (2), established during the proof of Prop. 1 in [3].

(iv) Write $U' = U \cup \{k\}$. We refer to Figure 1, where we show the logical multicast subtree spanned by k, U and their descendents, together with $a(U), a(U')$ and the root 0. From (i), $B(U')$ is the solution of the equation

$$\left(1 - \frac{\gamma(U')}{B(U')}\right) = \left(1 - \frac{\gamma(k)}{B(U')}\right) \prod_{j \in U} \left(1 - \frac{\gamma(j)}{B(U')}\right). \tag{32}$$

and $B(U)$ is the solution of

$$(1 - \gamma(U)/B(U)) = \prod_{j \in U} (1 - \gamma(j)/B(U)) \tag{33}$$

Now suppose that $B(U) \geq B(U')$. We shall show that this leads to a contradiction. Since then $B(U) \geq B(U') > \gamma(U')$, we can apply (i) and (ii) to (32) to obtain

$$\begin{aligned} 1 - \frac{\gamma(U')}{B(U)} &\geq \left(1 - \frac{\gamma(k)}{B(U)}\right) \prod_{j \in U} \left(1 - \frac{\gamma(j)}{B(U)}\right) \\ &= \left(1 - \frac{\gamma(k)}{B(U)}\right) \left(1 - \frac{\gamma(U)}{B(U)}\right), \end{aligned} \tag{34}$$

with the right-hand equality obtained by substitution of (33). Applying (2) at the node $a(U')$ we have

$$1 - \frac{\gamma(U')}{A(a(U'))} = \left(1 - \frac{\gamma(k)}{A(a(U'))}\right) \left(1 - \frac{\gamma(U)}{A(a(U'))}\right). \tag{35}$$

Since the assumption $B(U) \geq B(U')$ implies that $B(U) > \gamma(U')$, then comparing (34) with (35) and using (ii) again we find $A(a(U')) \leq B(U) = A(a(U))$. This is a contradiction since $a(U') \succ a(U)$ and T canonical implies $A(a(U')) > A(a(U))$. ■

While proving that DLT reconstructs the tree correctly, we find it useful to identify a subset S of V as a **stratum** if $\{R(k) : k \in S\}$ is a partition of R. If DLT works correctly, then before each execution of the while loop at line 4 of Figure 2, the set R' is a stratum and the set (V', L') of nodes and links is consistent with the actual tree (V, L) in the sense that it decomposes over subtrees rooted at the stratum R', i.e., $V' = \cup_{k \in R'} V(k)$ and $L' = \cup_{k \in R'} L(k)$. This is because any correct iteration of the loop that groups the children of node k has the effect

of joining subtrees rooted at nodes in $d(k)$, while modifying the partition $\{R(k) : k \in R'\}$ by replacing elements $\{R(j) : j \in d(k)\}$ by $R(k)$. The proof of Theorem 2 depends on the following Lemma that collects some properties of strata.

Lemma 3: If S is a stratum in a logical multicast tree (V, L) then

(i) If $k \in S$ then no ancestor or descendant of k lies in S.

(ii) Exactly one of the following alternatives applies to each non-root node k in V: (a) $k \in S$; (b) k has an ancestor in S; (c) k has at least two descendants in S.

Proof of Lemma 3: (i) If $j, k, \in S$ and $j \prec k$ then $R(j) \subset R(k)$, contradicting the partition property. (ii) If $k \notin S$, then there exists $j \in S$ obeying one of the alternatives $j \succ k$ or $j \prec k$, for otherwise $R(j)$ does not overlap with any element of the partition $\{R(j) : j \in S\}$. By (i), the alternatives are exclusive. There exists $j \in S$ with $j \succ k$, it is unique, by (i). If not, there exists $j \in S$ with $j \prec k$. In this case k cannot be a leaf node and hence $R(j) \subsetneq R(k)$ since k has branching ratio at least 2. Hence there must be at least one more node $j' \in S$ with $j \prec k$, since otherwise the partition $\{R(j) : j \in S\}$ would not cover R. ■

Proof of Theorem 2: (i) Suppose that DLT yields an incorrect tree, and consider the first execution of the loop during which (V', L') becomes inconsistent. Inconsistency could occur for the following reasons only:

1. *If the minimizing pair $\{u_1, u_2\}$ are not siblings.* Then there exists $t \prec a(u_1, u_2)$ that is the parent of either u_1 or u_2; say $t \succ u_1$. Since $u_1 \in R'$, by Lemma 3(i) no ancestor of u_1 – including t – can be in R'. Hence by Lemma 3(ii), there must be at least one node u_2' in addition to u_1 with the property that $u_2' \prec t$ and $u_2' \in R'$. Since the loss tree is canonical, $B(u_1, u_2') \leq A(t) < A(a(u_1, u_2)) = B(u_1, u_2)$, contradicting the minimality of $B(u_1, u_2)$. Hence the minimizing pair are siblings.

2. *If not all sibling nodes of u_1, u_2 are members of R'.* Let there be a sibling s of u_1 that is not in R'. Since $u_1 \in R'$, then by Lemma 3(i) no ancestor of u_1 – and hence no ancestor of its sibling s – can lie in R'. Since s itself is not in R', by Lemma 3(ii), there exist $s_1, s_2 \in R'$ with ancestor s. Since the loss tree is canonical, $B(s_1, s_2) \leq A(s) < A(a(u_1, u_2)) = B(u_1, u_2)$, contradicting the minimality of $B(u_1, u_2)$. Hence all siblings of u_1, u_2 are members of R'.

3. *If not all sibling nodes of u_1, u_2 are included in U of steps 5–7.* This would violate Prop 1(iii).

4. *If a node that is not a sibling of u_1, u_2 is included in U.* This would violate Prop 1(iv).

(ii) Since (i) allows the reconstruction of the loss tree from the outcome distribution, distinct loss trees can not give rise to the same outcome distributions, and hence the canonical loss tree is identifiable. ■

Proof of Theorem 3: Consider a maximal set $U = \{u_1, u_2, \ldots, u_m\}$ of siblings that is formed by execution of the while loop at line 6 in DLT; see Figure 2. We assume the non-trivial case that $m > 2$ and assume initially that U is unique. By Prop. 1(iii), $B(\cdot)$ is minimal within $R^{(0)} := R'$ on any pair of nodes from $S^{(0)} := U$. The action of DBLT can be described iteratively over $\ell = \{0, 1, \ldots, m\}$ as follows. After selecting $U^{(\ell)} = \{u_1^{(\ell)}, u_2^{(\ell)}\}$ in line 5, all pairs in $S^{(\ell+1)} = (S^{(\ell)} \setminus U^{(\ell)}) \cup \{U^{(\ell)}\}$ minimize $B(\cdot)$ over all pairs in $R^{(\ell+1)} = (R^{(\ell)} \setminus U^{(\ell)}) \cup \{U^{(\ell)}\}$ with the same minimum $B(U)$. This is because $(1 - \gamma(U^{(\ell)})/B(U)) = \prod_{u \in \mathcal{U}(\ell)}(1 - \gamma(u)/B(U))$ where $\mathcal{U}(\ell)$ denotes the members of U that are descended from $U^{(\ell)}$ in the binary tree built by DBLT. Hence $(1 - \gamma(U^{(\ell)})/B(U)) = (1 - \gamma(u_1^{(\ell)})/B(U))(1 - \gamma(u_2^{(\ell)})/B(U))$ and so $B(U^{(\ell)}) = B(U)$ by Prop. 1(i).

Thus for each step in DLT that groups the nodes in U, there are $m - 1$ steps of DBLT that successively group the same set of nodes. Since $B(U^{(\ell)}) = B(U)$ for all ℓ, each node j added in DBLT has $\alpha_j = 1$, apart from the last one. Therefore, TP(0) acts to excise all links between the binary nodes $U^{(0)}, \ldots, U^{(m)} - 1$. Thus DLT $=$ TP(0) $\circ$ DBLT. If U in not unique, the same arguments apply, except now there can be alternation of grouping operations acting on different maximal sibling sets. ■

Proof of Theorem 4: Since each $\widehat{\gamma}(U)$ is the mean of n independent random variables then by the Strong Law of Large Numbers, $\widehat{\gamma}(U)$ converges to $\mathsf{E}[\widehat{\gamma}(U)] = \gamma(U)$ almost surely as $n \to \infty$. In Theorem 1 of [3] it is shown that $B(U)$ is a continuous function of $\{\gamma(a(U)), \{\gamma(k) : k \in U\}\}$, from which the result follows. ■

Proof of Theorem 5: Let U denote a generic binary subset of R' that minimizes $B(\cdot)$ when DBLT is applied to $(\mathcal{T}, \alpha)$. Assume initially that the minimizing U is unique. Since the loss tree is canonical, $B(U) < B(U')$ for any other candidate binary set U'; by the convergence property of Theorem 4, $\widehat{B}(U) < \widehat{B}(U')$ for all n sufficiently large. Hence the nodes in U are grouped correctly.

But it may happen, by coincidence, that the minimizing U is not unique. Then there are pairs $U^{(1)}, \ldots, U^{(m)}$ that minimize B. Since the tree is canonical, then after each $U^{(\ell)} \in R'$ has been grouped, the remaining pairs are still minimizers of $B(\cdot)$ amongst all pairs of the reduced set $(R' \setminus U^{(\ell)}) \cup \{U^{(\ell)}\}$ in line 10 of Figure 4. Hence DBLT picks these pairs successively for grouping until all pairs have been picked.

With BLT, the $\widehat{B}(U^{(\ell)})$ are no longer equal. But for sufficiently large n they will still all be less than $\widehat{B}(U')$ for any other candidate pair U', by Theorem 4. Thus BLT will successively group the pairs $U^{(1)}, \ldots, U^{(m)}$ in some

random order that depends on the relative magnitude of the $\widehat{B}(U^{(\ell)})$. But the order is not important, since the end result is just to have the pairs formed as DBLT would have. ■

Proof of Theorem 8: It suffices to show that $\lim_{n\to\infty} \mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\rm ML} = \mathcal{T}') = 0$ for each $\mathcal{T}' \neq \mathcal{T}$, Let $p(x;\mathcal{T},\alpha)$ denote the probability of the outcome $x \in \{0,1\}^R$ under the loss tree $(\mathcal{T},\alpha)$. Under our assumptions, if $\mathcal{T}' \neq \mathcal{T}$, the Kullback-Leibler information

$$I\left((\mathcal{T},\alpha),(\mathcal{T}',\alpha')\right) := \mathsf{E}_{\mathcal{T},\alpha}(\log(p(X;\mathcal{T},\alpha)/p(X;\mathcal{T}',\alpha'))) \tag{36}$$

is a continuous function of $\alpha' \in \mathcal{A}^{\varepsilon}_{\mathcal{T}'}$, and is strictly positive because of identifiability. Thus there is a number $\delta > 0$ such that $I\left((\mathcal{T},\alpha),(\mathcal{T}',\alpha')\right) \geq \delta$ for all $\alpha' \in \mathcal{A}^{\varepsilon}_{\mathcal{T}'}$. Now

$$\mathsf{P}_{\mathcal{T},\alpha}(\widehat{\mathcal{T}}_{\rm ML} = \mathcal{T}') \leq \tag{37}$$
$$\mathsf{P}_{\mathcal{T},\alpha}\left(\cup_{\alpha'\in\mathcal{A}^{\varepsilon}_{\mathcal{T}'}}\left\{\frac{1}{n}\sum_1^n \log\frac{p(X^{(i)};\mathcal{T}',\alpha')}{p(X^{(i)};\mathcal{T},\alpha)} \geq 0\right\}\right).$$

Since $\alpha' \in \mathcal{A}^{\varepsilon}_{\mathcal{T}'}$, the density $p(x;\mathcal{T}',\alpha')$ is bounded away from zero, hence the conditions of Jennrich's [8] uniform strong law of large numbers are satisfied. Thus, $\mathsf{P}_{\mathcal{T},\alpha}$-almost surely,

$$\frac{1}{n}\sum_1^n \log\frac{p(X^{(i)};\mathcal{T}',\alpha')}{p(X^{(i)};\mathcal{T},\alpha)} \longrightarrow -I\left((\mathcal{T},\alpha),(\mathcal{T}',\alpha')\right) \leq -\delta \tag{38}$$

uniformly in $\alpha' \in \mathcal{A}^{\varepsilon}_{\mathcal{T}'}$, whence the RHS of (37) converges to zero as $n \to \infty$. ■

Proof of Theorem 9: Recall from the proof of Theorem 8 that the Kullback-Leibler information $I(\theta,\theta')$ is a continuous function of θ', and, because of identifiability, has a unique minimum, namely 0, at $\theta' = \theta$. Given any neighborhood U of $\theta \in \Theta$, it follows that, for sufficiently small $\varepsilon > 0$, the set $C_\varepsilon = \{\theta' : I(\theta,\theta') < \varepsilon\}$ is contained in U. Using Theorem 7.80 of Schervish [17], we have, for $n \to \infty$,

$$\pi(U|x) \longrightarrow 1,\ \mathsf{P}_\theta - a.s. \tag{39}$$

Consider the pseudo-Bayes classifier $\widehat{\mathcal{T}}_\pi$, which now takes the form

$$\widehat{\mathcal{T}}_\pi(x) = \arg\max_{\tau\in\mathcal{T}(R)} \pi(\tau \times \mathcal{A}^0_\tau|x). \tag{40}$$

From (39) we obtain that, $\mathsf{P}_{\mathcal{T},\alpha}$ almost surely, $\pi(\{\mathcal{T}\}\times\mathcal{A}^0_{\mathcal{T}}|x) \longrightarrow 1$ and $\pi(\{\tau\}\times A_\tau|x) \longrightarrow 0$ for $\tau \neq \mathcal{T}$, whence $\widehat{\mathcal{T}}_\pi(x) = \mathcal{T}$ for sufficiently large n, $\mathsf{P}_{\mathcal{T},\alpha}$ almost surely. ■

Proof of Lemma 1: Assume that a number of groupings have been formed, after which k_1, k_2 are candidate nodes descended from i, while k_3 is some other candidate node not descended from i. Since the grouping thus far is correct, k_3 cannot be i or an ancestor of i, and hence $R(k_3) = S_3 \subset R_{\mathcal{T}} \setminus R_{\mathcal{T}}(i)$. Let $S_j = R_{\mathcal{T}}(k_j)$, $j = 1,2$. All the S_j are disjoint. By arguments similar to those used in the proof of Theorem 2, $B(\{k_1,k_3\}) = B(S_1,S_3) > B(S_1,S_2) = B(\{k_1,k_2\})$. Thus correct grouping of k_1, k_2 by BLT is guaranteed if (16) holds for all $(S_1,S_2,S_3) \in \mathcal{S}(i)$. ■

Proof of Theorem 10: Since for each $S \subseteq R$, $\widehat{\gamma}(S)$ is the mean of i.i.d. random variables $\widehat{Y}^{(i)}_S$, the variables $\sqrt{n}\cdot(\widehat{\gamma}-\gamma)$, $\widehat{\gamma} = \{\widehat{\gamma}(S)\}_{S\subseteq R}$, converge to a multivariate Gaussian random variable as $n \to \infty$. Since $\widehat{D}$ is a differentiable function $\mathcal{D}$ of $\widehat{\gamma}$, the Delta method insures that the stated convergence holds.

To prove (i) observe that since $a(S_1), a(S_3) \prec a(S_1 \cup S_3)$ then $B(S_1,S_3) = A(a(S_1 \cup S_3))$. Since S_1 and S_2 may not satisfy $a(S_1), a(S_2) \prec a(S_1 \cup S_2)$–this may occur whenever there was a grouping error in any of the steps that lead to the construction of node S_1 and/or node S_2–we need to explicitly write the expression for $B(S_1,S_2)$,

$$\begin{aligned} B(S_1,S_2) &= \frac{\mathsf{P}[\vee_{j\in S_1} X_j = 1]\mathsf{P}[\vee_{j\in S_2} X_j = 1]}{\mathsf{P}[\vee_{j\in S_1} X_j \cdot \vee_{j\in S_2} X_j = 1]} \\ &= \mathsf{P}[X_{a(S_1\cup S_2)} = 1] \\ &\times \frac{\mathsf{P}[\vee_{j\in S_1} X_j = 1 | X_{a(S_1\cup S_2)} = 1]}{\mathsf{P}[\vee_{j\in S_1} X_j = 1 | X_{a(S_1\cup S_2)} = 1,\ \vee_{j\in S_2} X_j = 1]} \\ &= A(a(S_1\cup S_2))\psi_{(S_1,S_2)} \end{aligned} \tag{41}$$

where $\psi_{(S_1,S_2)} := \frac{\mathsf{P}[\vee_{j\in S_1} X_j = 1 | X_{a(S_1\cup S_2)}=1]}{\mathsf{P}[\vee_{j\in S_1} X_j = 1 | X_{a(S_1\cup S_2)}=1,\ \vee_{j\in S_2} X_j = 1]}$. Observe from Proposition 1(iv) that $\psi_{(S_1,S_2)} \leq 1$. Intuitively, the smaller $\psi_{(S_1,S_2)}$, the greater the error committed so far in classifying the subtree rooted at i. (i) then follows as for $\|\overline{\alpha}\| \to 0$ it is easy to verify that $A(k) = 1 - s(k) + O(\|\overline{\alpha}\|^2)$ and $\chi_{(S_1,S_2)} = 1 - O(\|\overline{\alpha}\|^2)$. To prove (ii), a standard application of the Delta method shows that the collection of $\sqrt{n}(\widehat{B}(S_1,S_2) - B(S_1,S_2))$ converge as $n \to \infty$ to a multivariate Gaussian random variable with mean zero and covariance matrix

$$\nu_{(S_1,S_2),(S_3,S_4)} = \sum_{S,S'\subseteq R} \frac{\partial B(S_1,S_2)}{\partial\gamma(S)}\, C_{S,S'} \frac{\partial B(S_3,S_4)}{\partial\gamma(S')}. \tag{42}$$

where $C_{S,S'} = \mathsf{Cov}[\widehat{Y}^{(i)}_S, \widehat{Y}^{(i)}_{S'}]$. Now, following the same lines of Theorem 5 in [3], we have that $C_{S,S'} = s(a(S\cup S')) + O(\|\overline{\alpha}\|^2)$, and $\frac{\partial B(S_1,S_2)}{\partial\gamma(S)} = \delta_{(S_1\cup S_2),S} + O(\|\overline{\alpha}\|^2)$ by direct differentiation. Therefore, we have $\nu_{(S_1,S_2),(S_3,S_4)} = C_{(S_1\cup S_2),(S_3\cup S_4)} + O(\|\overline{\alpha}\|^2)$. Hence,

$$\begin{aligned} \sigma^2(S_1,S_2,S_3) &= \nu_{(S_1,S_3),(S_1,S_3)} + \nu_{(S_1,S_2),(S_1,S_2)} \\ &\quad -2\nu_{(S_1,S_2),(S_1,S_3)} + O(\|\overline{\alpha}\|^2) \end{aligned} \tag{43}$$

$$= \quad s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3)) + O(\|\overline{\alpha}\|^2)$$

Finally, (iii) follows as $s(a(S_1 \cup S_2)) - s(a(S_1 \cup S_3))$ is minimized when $a(S_1 \cup S_2) = i$ and $a(S_1 \cup S_3) = f(i)$. ■

REFERENCES

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior, IEEE Communications Magazine, 38(5), 152–159, May 2000.

[2] J. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed., Springer, 1985.

[3] R. Caceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics", IEEE Trans. on Information Theory, 45: 2462–2480, 1999.

[4] R. Caceres, N.G. Duffield, J.Horowitz F. Lo Presti and D. Towsley, "Statistical Inference of Multicast Network Topology", in Proc. 1999 IEEE Conf. on Decision and Control, Phoenix, AZ, Dec. 1999.

[5] A. Dembo and O. Zeitouni, *Large Deviation Techniques and Applications.* Jones and Bartlett, Boston-London, 1993.

[6] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", in Proc. IEEE Infocom 2000, Tel Aviv, March 2000.

[7] M. Handley, S. Floyd, B. Whetten, R. Kermode, L. Vicisano, M. Luby. "The Reliable Multicast Design Space for Bulk Data Transfer," RFC 2887, Internet Engineering Task Force, Aug. 2000.

[8] R. Jennrich, "Asymptotic properties of nonlinear least-squares estimators", *Ann. Math. Stat.* 40:633–643, 1969.

[9] B.N. Levine, David Lavo , and J.J. Garcia-Luna-Aceves, "The Case for Concurrent Reliable Multicasting Using Shared Ack Trees," Proc. ACM Multimedia Boston, MA, November 18–22, 1996.

[10] B.N. Levine, S. Paul, J.J. Garcia-Luna-Aceves, "Organizing multicast receivers deterministically according to packet-loss correlation," Proc. ACM Multimedia 98, Bristol, UK, September 1998.

[11] F. Lo Presti, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[12] `mtrace` – Print multicast path from a source to a receiver. See ftp://ftp.parc.xerox.com/pub/net-research/ipmulti

[13] `ns` – Network Simulator. See http://www-mash.cs.berkeley.edu/ns/ns.html

[14] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," IEEE Communications Magazine, Vol. 36, No. 8, pp. 48–54, August 1998.

[15] M.J.D. Powell, "Gradient conditions and Lagrange multipliers in nonlinear programming". Lecture 9 in L.C.W. Dixon, E. Spedicato, G.P. Szegö (eds.), "Nonlinear optimization: theory and algorithms", Birkhäuser, 1980, p. 210

[16] S. Ratnasamy & S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", in Proc. IEEE Infocom'99, New York, March 1999

[17] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[18] R.J. Vanderbei and J. Iannone, "An EM approach to OD matrix estimation," Technical Report, Princeton University, 1994

[19] Y. Vardi, "Network Tomography: estimating source-destination traffic intensities from link data," J. Am. Statist. Assoc., 91: 365–377, 1996.

[20] B. Whetten, G. Taskale. "Reliable Multicast Transport Protocol II," IEEE Networks, 14(1), 37–47, Jan./Feb. 2000.

Multicast Topology Inference from End-to-end Measurements*

N.G. Duffield[‡,†] J. Horowitz[♠] F. Lo Presti[‡,§] D. Towsley[§]

[‡]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
{duffield,lopresti}@research.att.com

[♠]Dept. Math. & Statistics
University of Massachusetts
Amherst, MA 01003, USA
joeh@math.umass.edu

[§]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@gaia.cs.umass.edu

Abstract

This paper describes a class of statistical estimators of multicast tree topology based on end-to-end multicast traffic measurements. This approach allows the determination of the logical multicast topology without assistance from the underlying network nodes. We provide five instances of the class, variously using loss or delay measurements. We compare their accuracy and computational cost, and recommend the best choice in each of the light and heavy traffic load regimes.

1 Introduction and Motivation

The use of multicast shows great promise for determining internal network characteristics based solely on end-to-end measurements. This is because multicast introduces correlation in the end-to-end behavior observed by different receivers within the same multicast session. This correlation can be used to estimate packet loss rates, [1], packet delay distributions, [5], and packet delay variances, [4]. These methods can be used as part of a multicast–capable measurement infrastructure, such as NIMI (National Internet Measurement Infrastructure) [9], for the purpose of monitoring internal network behavior.

All of these multicast-based methods can be applied to end-to-end multicast observations made of packets traversing a fixed, but arbitrary, topology. Knowledge of the multicast topology is required in order to apply the methods. Unfortunately, knowledge of the tree topology is not always available. This motivates the need for algorithms that can identify the topology of the multicast tree. Another motivation is that knowledge of the multicast topology can be of use to multicast applications. For example several reliable multicast protocols (e.g., RMTP [8]) rely on logical hierarchies based on the underlying topology if possible. Other applications attempt to group receivers that share the same network bottleneck, [10].

In this paper, we present a general framework within which to develop algorithms for identifying the multicast topology based on end-to-end observations. Once the topology has been identified, any of the methods mentioned above for identifying internal network behavior can then be applied. The development of an algorithm is based on the presence of a packet performance measure that monotonically increases as the packet traverses down the tree and that can be estimated based on observations made at the receivers. We provide additional constraints on the measures such that the resulting algorithm can be shown to be asymptotically consistent, i.e., it identifies the correct topology almost surely as the number of observations goes to infinity. Examples of performance measures that yield such algorithms include loss rate, delay variance, average delay, and link utilization.

Several algorithms have been proposed for identifying multicast topologies based on loss observations made at receivers. For example, [10] presented an algorithm for identifying a multicast tree when it is a binary tree. [2] established the correctness of this al-

*This work was supported in part by DARPA and the AFL under agreement F30602-98-2-0238

†Corresponding Author: Tel. +1 973 370 8726; Fax +1 973 360 8050

gorithm and introduced several other loss-based algorithms for identifying general trees. They concluded that an algorithm that constructs a binary tree and subsequently prunes branches whose loss rates are close to zero was best suited for topology identification. The framework presented in this paper comes from the recognition that this last approach can be applied to observations of other end-to-end measures such as delays.

Topology discovery is an essential part of several current measurement infrastructure projects, including CAIDA, Felix, IPMA, NIMI and Surveyor; see [3]. We contrast our approach with that of the commonly used diagnostic tools `traceroute` and `mtrace` [6] that discover physical topology. These require cooperation from intervening nodes (in the generation of ICMP messages, or in maintaining counters) and their widespread use raises issues of scaling in topologies with many leaves. The present methods complement these, being able to discover logical multicast topology and it changes without cooperation from the network, using measurement traffic whose volumes that scale well for larger topologies; see [2] for further discussion.

The paper is organized as follows. We introduce our framework in Section 2 and provide conditions under which the resulting algorithms are strongly consistent. Applications to loss and delay measures are presented in Section 3. In Section 4 we analyze the probability of topology misclassification, asymptotically for large numbers of probes. Section 5 reports on a simulation study on the effectiveness of different algorithms obtained through this approach and makes recommendations as to when they perform well. Section 6 concludes the paper.

2 A Framework for Topology Inference

Tree Model. The physical multicast tree comprises actual network elements (the nodes), and the communication links than join them. When a packet is multicast to a set of receivers, only one copy of the packet traverses each link of the tree; copies are created at the branch points of the tree, one per outgoing link. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except the leaf nodes and possibly the root, must have 2 or more children.

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify the root node 0 as the source of probes, and $R \subset V$ as the set of leaf nodes (identified as the set of receivers). The set of children of node $j \in V$ is denoted by $d(j)$. Each node, k, apart from the root has a parent $f(k)$ such that $(f(k), k) \in L$. Define recursively the compositions $f^n = f \circ f^{n-1}$ with $f^1 = f$. We will sometimes refer to the link $(f(k), k)$ as simply link k. Nodes are said to be siblings if they have the same parent. If $k = f^m(j)$ for some $m \in \mathbb{N}$ we say that j is descended for k (or equivalently that k is an ancestor of j) and write the corresponding partial order in V as $j \prec k$. $a(i, j)$ will denote the minimal common ancestor of i and j in the $\preceq$-ordering. For $k \in V$ we let $\mathcal{T}(k) = (V(k), L(k))$ denote the subtree of $\mathcal{T}$ that is rooted at k, and set $R(k) = R \cap V(k)$.

Tree Marks. The experience of a multicast packet on its passage down the tree is modeled by a random process of **marks** $x = (x_k)_{k \in V}$. Each mark x_k takes a value in a set $\mathcal{X}$ appropriate to the problem of interest. x_k specifies the experience of a packet traversing link k. In the setting of packet loss, for example, we take $\mathcal{X} = \{0, 1\}$, where $x_k = 1$ indicates that a packet present at node $f(k)$ is successfully transmitted to node k, while $x_k = 0$ indicates that it would be lost.

Composing Marks Along Paths. A path is a set of contiguous links, identified by the ordered set of link endpoints $(k_1, \ldots, k_\ell)$ where $k_i = f(k_{i+1})$. We will sometimes use the notation $\boldsymbol{p}(k)$ to denote the path $(k_1, \ldots, k_\ell)$ where $k_1 = 0$ and $k_\ell = k$. The experience of the multicast packet on a path is obtained by composing the marks from each link on the path to form a mark for the path. *Composition* is an associative and commutative binary operation $\otimes$ on $\mathcal{X}$. A path $\boldsymbol{p} = (k_1, \ldots, k_\ell)$ has mark $x_{\boldsymbol{p}}$ formed by successively composing the marks of its constituent links: $x_{\boldsymbol{p}} = x_{k_1} \otimes \ldots \otimes x_{k_\ell}$. We assume that $\mathcal{X}$ contains an identity $\boldsymbol{z}$ such that $\boldsymbol{z} \otimes x = x$ for all $x \in \mathcal{X}$.

In the example of loss we can compose link marks using the minimum $x_1 \otimes x_2 = x_1 \wedge x_2$. This models the physical property the loss occurs on a path if it occurs for any link on the path. The identity is $\boldsymbol{z} = 1$.

Measurements and Marks. We assume that the individual marks x_k are not directly knowable. Rather, end-to-end measurements comprise the marks $x_{\boldsymbol{p}(k)}$ along paths terminating at leaf nodes $k \in R$. Our task will be to infer the underlying topology from these path marks alone. This information can also be used to to characterize the individual links, by inferring the distribution of their link marks.

Mark Aggregation. We also equip $\mathcal{X}$ with an aggregation operation that summarizes the experience of packets over a set of possibly intersecting paths. We restrict attention to binary trees. *Aggregation* is then a binary operation $\oplus$ on $\mathcal{X}$. The multicasting property is reflected in composition $\otimes$ being distributive over aggregation $\oplus$, i.e.,

$$(x_1 \otimes x_2) \oplus (x_1 \otimes x_3) = x_1 \otimes (x_2 \oplus x_3) \qquad (1)$$

for any $x_1, x_2, x_3 \in \mathcal{X}$. The reason that this relation holds becomes clearer if we consider a four-node logical multicast tree with root 0 having a single child 1, the latter node having children $2, 3$ that are leaves. Eq. (1) says that when calculating the aggregate mark for intersecting paths $(1, 2)$ and $(1, 3)$, we can factor out the common mark on the common link 1. When dealing with loss, for example, we will take aggregation as maximization, i.e. $x_1 \oplus x_2 = x_1 \vee x_2$.

Aggregating Receiver Marks. Nodes $k \in V \setminus (R \cup \{0\})$ have two children, which we denote by $h(k)$ and $h^*(k)$. For each $k \in V \setminus \{0\}$ we define the aggregate marks over the paths to all receivers descended from k recursively through

$$\widetilde{x}_k = \begin{cases} x_{\boldsymbol{p}(k)} & k \in R \\ \widetilde{x}_{h(k)} \oplus \widetilde{x}_{h^*(k)} & \text{otherwise} \end{cases} . \qquad (2)$$

When $\oplus$ is associative (i.e. $(m_1 \oplus m_2) \oplus m_3 = m_1 \oplus (m_2 \oplus m_3)$ we can write $\widetilde{x}_k = \oplus_{j \in R(k)} x_{\boldsymbol{p}(j)}$. This is the case for loss, where $\widetilde{x}_k$ is 1 if the packet reaches some receiver descended from k and 0 otherwise. However, we have one example where $\oplus$ is not associative.

Mark Distributions. We assume that the marks are independently distributed according to a mark distribution $\omega = (\omega_k)_{k \in V}$, where ω_k is the distribution of the mark x_k. The distribution $\omega_{\boldsymbol{p}}$ of the composite mark of a path $\boldsymbol{p} = (k_1, \ldots, k_\ell)$ is determined by convolution in the usual way: for a measurable subset B of $\mathcal{X}$, $\omega_{\boldsymbol{p}}(B) = (\omega_{k_1} * \ldots * \omega_{k_\ell})(B) := \int_{x_1 \otimes \ldots \otimes x_\ell \in B} \omega_{k_1}(dx_1) \ldots \omega_{k_\ell}(dx_\ell)$. The joint distribution of $\widetilde{x}_{k_1}, \ldots, \widetilde{x}_{k_\ell}$ will be denoted $\widetilde{\omega}_{k_1,\ldots,k_\ell}$.

Deterministic Reconstruction of Binary Trees. The classification of trees relies on being able to identify certain characteristics of paths that do not terminate at leaves from the characteristics of those that do. Such a characteristic ϕ will be termed *estimable*; the precise definition is below. We seek estimable characteristics that *increase* as paths lengthen; this allows us to select as siblings those nodes for which the characteristic ϕ on the common portion of the path from 0 is maximized.

Let $\mathcal{T} = (V, L)$ be a binary tree. Let Ω be a set of probability distributions on $\mathcal{X}$ that is closed under convolution, and denote by Ω^V the corresponding product distribution of marks on $\mathcal{T}$. Let ϕ be a weakly continuous functional the set of measures on $\mathcal{X}$ that takes values in some linear space $\mathcal{Q}$. We equip $\mathcal{Q}$ with the usual partial order, i.e., $(q_i) > (q_i')$ in $\mathcal{Q}$ iff $q_i > q_i'$ for all i. Let $\delta_{\boldsymbol{z}}$ denote the distribution which has unit mass at the composition identity $\boldsymbol{z}$.

Definition 1 *(i) ϕ is called* **estimable** *if there exists a function Φ such that, for each $\omega \in \Omega^V$, and $j, k \in V$ with $a(j,k) \neq j, k$, $\phi(\omega_{\boldsymbol{p}(a(j,k))}) = \Phi(\widetilde{\omega}_{j,k})$.*

(ii) ϕ is called **increasing** *if, $\phi(\delta_{\boldsymbol{z}}) = 0 \in \mathcal{Q}$, and for all $\omega \in \Omega$, $\phi(\omega_{\boldsymbol{p}}) < \phi(\omega_{\boldsymbol{q}})$ when $\boldsymbol{p}$ is a proper subpath of $\boldsymbol{q}$.*

The condition $\phi(\delta_{\boldsymbol{z}}) = 0$ says that a link whose marks never change the marks of paths traversing it, do not increase the value of ϕ.

Given an estimable, increasing ϕ, and a distribution ω, write $\Phi_{j,k} = \Phi(\widetilde{\omega}_{j,k})$. The topology can

1. *Input*: The set of receivers $R = \{i_1, \ldots, i_r\}$ and the leaf mark distributions $\widetilde{\omega}_{i_1,\ldots,i_r}$.
2. $R' := R$; $V' := R'$; $L' = \emptyset$;
3. **while** $|R'| > 1$ **do**
4. **select** $U = \{j, k\} \subseteq R'$ with maximal $\Phi_{j,k}$;
5. $V' := V' \cup \{U\}$;
6. $L' = L' \cup \{(U, \ell) : \ell \in U\}$;
7. $R' := (R' \setminus U) \cup \{U\}$;
8. **enddo**
9. **if** $\Phi_{j,k} > 0$ **do**
10. $V' := V' \cup \{0\}$; $L' = L' \cup \{(0, R')\}$;
11. **enddo**
12. *Output*: tree (V', L') ;

Figure 1: Deterministic Binary Tree Classification Algorithm (DBT).

be reconstructed from ϕ as follows. The key observation from (ii) is that $\Phi_{j,k} > \Phi_{j',k'}$ whenever $a(j,k) \prec a(j',k')$. Thus $\Phi_{j,k}$ is maximized when j, k are siblings in R. For if not, then one of the receivers, say j, would have a sibling k' for which $\Phi_{j,k'} > \Phi_{j,k}$. Thus the siblings can be identified on the basis of leaf distributions alone. Substituting a composite node that represents their parent and iterating, should then reconstruct the binary tree. This approach is formalized in the Deterministic Binary Tree Classification Algorithm (DBT); see Figure 1.

DBT operates as follows. R' denotes the current set of nodes from which a pair of siblings will be chosen, initially equal to the receiver set R. We first find the pair $U = \{j, k\}$ that maximizes $\Phi_{j,k}$ (line 4). This identifies the members of U as siblings, and the set U is used to represent their parent. Correspondingly, we add j, k to the list V' of nodes (line 5), we add $(U, j), (U, k)$ to the list L' of links (line 6), and replace j and k by U in the set R' of nodes available for pairing in the next stage (line 7). This process is repeated until all sibling pairs have been identified (loop from line 3). Finally, we test in line 9 whether the last node grouped should be taken as the root node. If the last node identified were not the root node, equality in the test would contradict the increasing property of ϕ. Otherwise, we adjoin a root node, and a link joining it to its single descendant (line 10).

We say that DBT reconstructs the binary logical multicast tree (V, L) if given the receiver set R and the leaf mark distribution $\omega_{\boldsymbol{p}(R)}$, it produces (V, L) as its output. Clearly this happens if and only if before each iteration of the while loop 3 in Figure 1, (V', L') can be decomposed in terms of disjoint subtrees $V' = \sum_{k \in R'} V(k)$ and $L' = \sum_{k \in R'} L(k)$. These subtrees may just be trivial ones $\mathcal{T}(k) = (\{k\}, \emptyset)$ comprising a root node k. We note also that these trees cover R, i.e. $R = \cup_{k \in R'} R(k)$. These properties hold before the first while loop, and hold subsequently since each loop of a successful reconstruction amalgamates binary subtrees rooted at siblings.

Theorem 1 *Let $\mathcal{T}$ be a binary tree, equipped with an estimable and increasing function ϕ. Then* DBT *reconstructs $\mathcal{T}$.*

Proof of Theorem 1: Suppose the algorithm does not reconstruct the tree. Then there must be an iteration of the while loop for which j and k in line 4 of Figure 1 are not siblings. Consider R', V' at the start of the first loop that this occurs. Let ℓ be the sibling of j. $\ell \notin R'$ since $a(j,\ell) \prec a(j,k)$ implies $\Phi_{j,\ell} > \Phi_{j,k}$, contradicting the maximality of $\Phi_{j,k}$. Since the subtrees comprising (V', L') are disjoint, no ancestor of j (or hence of ℓ) can lie in R'. Since the tree is binary, ℓ must have at least two descendents t_1, t_2 in R' since otherwise $\cup_{r \in R'} R(r)$ would not cover R. Since $a(t_1, t_2) \prec \ell$, then $\Phi_{t_1,t_2} > \phi(\omega_{\boldsymbol{p}(\ell)}) > \phi(\omega_{\boldsymbol{p}(a(j,k))}) = \Phi_{j,k}$, contradicting the maximality of $\Phi_{j,k}$. ∎

Reconstruction of Binary Trees from Measurements. Now we switch to the context that a stream of probes is dispatched from the source, each giving rise to an independent realization of the mark process. Let $x^{(i)}$ denote the i^{th} such realization. Each realization gives rise to a set of measurements $\{x^{(i)}_{\boldsymbol{p}(k)} : k \in R\}$ at the leaves. Suppose that some subtree $(V(k), L(k))$ of the tree is already identified. Then we can aggregate the measured leaf marks analogously to (2), defining $\widetilde{x}^{(i)}_k = x^{(i)}_{\boldsymbol{p}(k)}$ for $k \in R$, and forming $\widetilde{x}^{(i)}_k = \widetilde{x}^{(i)}_{h(k)} \oplus \widetilde{x}^{(i)}_{h^*(k)}$ by recursion for $k \notin R$.

Let $\widetilde{\omega}_k^{(n)} = n^{-1}\sum_{i=1}^n \delta_{\widetilde{x}_k^{(i)}}$ denote the empirical distribution of $\widetilde{x}_k^{(n)}$; here δ_y denotes the unit mass at $y \in \mathcal{X}$. We estimate $\mathcal{T}$ by the topology $\mathcal{T}^{(n)}$ obtained by using the $\widetilde{\omega}^{(n)}$ in place of $\widetilde{\omega}$ in the DBT algorithm. Specifically, we use $\Phi_{j,k}^{(n)} := \Phi(\widetilde{\omega}_{j,k}^{(n)})$ in place of $\Phi_{j,k}$ in line 3 of Figure 1. We call the resulting algorithm the Binary Tree Classification Algorithm (BT).

Theorem 2 *Under the conditions of Theorem 1, with probability* 1, $\mathcal{T}^{(n)} = \mathcal{T}$ *for sufficiently large* n. *Hence* $\mathcal{T}^{(n)}$ *is a consistent estimator of* $\mathcal{T}$ *and* $\lim_{n\to\infty} \mathsf{P}_\omega[\mathcal{T}^{(n)} \neq \mathcal{T}] = 0$.

Proof of Theorem 2: By the law of large numbers, $\omega^{(n)}$ converges weakly to ω, almost surely. Since ϕ is weakly continuous each $\Phi_{j,k}^{(n)}$ converges almost surely to $\Phi_{j,k}$. Then, almost surely, for all sufficiently large n, the relative ordering of the $\Phi_{j,k}^{(n)}$ is the same as that of the $\Phi_{j,k}$ for pairs j,k for which the $\Phi_{j,k}$ are distinct. Hence BT reconstructs the tree in the same manner as DBT, except possibly varying the order of grouping amongst sets of sibling pairs (j,k) with identical $\Phi_{j,k}$. The last two statements then follow by standard results. ∎

Characterizing Link Behavior. In many of the examples of the next section ϕ is additive over links. i.e. $\phi(\omega_1 * \omega_2) = \phi(\omega_1) + \phi(\omega_2)$. Then we can ascribe a descriptor ϕ_k, such as a packet loss rate, to each link $(f(k),k)$ through $\phi_k = \phi(\omega_{\boldsymbol{p}(k)}) - \phi(\omega_{\boldsymbol{p}(f(k))})$. (This may conveniently be done during the execution of DBT or BT).

Extension to General Trees. Inference of general trees is accomplished as follows. For simplicity assume that ϕ is additive. Then application of DBT to an arbitrary tree results in a binary tree that contains fictitious links k such that $\phi_k = 0$. The tree can then be pruned by removing any such link and identifying its endpoints. It can be shown that this procedure yields the true general tree. In BT it is necessary to apply a threshold $\varepsilon > 0$ and prune all links k with $\phi_k \leq \varepsilon$. This is because for finitely many probes, statistical fluctuations lead the characteristic ϕ of the fictitious links to differ slightly from zero. It can be shown that for sufficiently many probes, this approach reconstructs any general tree for which all $\phi_k > \varepsilon$.

3 Instances of Topology Inference

In this section we specify instances of the framework described above, specifying the setting (the marks $\mathcal{X}$, etc) and the the forms of the functions ϕ, Φ and $\Phi^{(n)}$. Theorem 1 and 2 then apply immediately in each case.

3.1 Loss-Based Inference

In this case ϕ is (a function of) the probability of successful transmission from the root to a given node. In the above formalism, we take $\mathcal{X} = \{0,1\}$, where 0 indicates packet loss and 1 transmission. Composition is by taking minima $x_1 \otimes x_2 = x_1 \wedge x_2$ and the identity is $\boldsymbol{z} = 1$; a packet is transmitted on a path if it would be transmitted on all links of that path. Aggregation is by taking maxima $x_1 \oplus x_2 = x_1 \vee x_2$; hence $\widetilde{x}_k = 1$ if a packet reaches any receiver descended from k. It can be shown [1] that

$$\mathsf{P}_\omega[x_{\boldsymbol{p}(a(j,k))} = 1] = \frac{\mathsf{P}_\omega[\widetilde{x}_j = 1]\mathsf{P}_\omega[\widetilde{x}_k = 1]}{\mathsf{P}_\omega[\widetilde{x}_j = \widetilde{x}_k = 1]}. \quad (3)$$

Using $\mathcal{Q} = \mathbb{R}_+$, we define ϕ to act on the generic measure on $\mathcal{X}$ as $\phi((1-\alpha)\delta_0 + \alpha\delta_1) = -\log(\alpha)$ for $\alpha \in [0,1]$. Clearly $\phi(\delta_{\boldsymbol{z}} = 0)$. We write $\Phi_{j,k} = \log \mathsf{P}_\omega[\widetilde{x}_j = \widetilde{x}_k = 1] - \log \mathsf{P}_\omega[\widetilde{x}_j = 1] - \log \mathsf{P}_\omega[\widetilde{x}_k = 1]$. ϕ is additive over links, and the link characteristic is $\phi_k = -\log \mathsf{P}_{\omega_k}[x_k = 1]$, i.e., the negative log probability of successful transmission over link k. Thus ϕ is increasing provided the link loss probabilities are strictly positive. For inference from measurements, $\Phi_{j,k}^{(n)} = \log n + \log(\sum_{i=1}^n \widetilde{x}_j^{(i)}\widetilde{x}_k^{(i)}) - \log(\sum_{i=1}^n \widetilde{x}_j^{(i)}) - \log(\sum_{i=1}^n \widetilde{x}_k^{(i)})$ where we have expressed the various probabilities in terms of the empirical means.

3.2 Delay Covariance-Based Inference

In this case the increasing function ϕ is the variance of the cumulative delay from the root to a given node. In the formalism, $\mathcal{X} = \mathbb{R}$, with x_k the delay encountered on link k. (The formalism extends to loss by

using $x_k = \infty$ to denote loss; we treat this elsewhere [4]). Composition adds delays along a path: $x_1 \otimes x_2 = x_1 + x_2$. The identity is $\boldsymbol{z} = 0$. Aggregation takes the mean of two delays $x_1 \oplus x_2 = (x_1 + x_2)/2$. Taking $\mathcal{Q} = \mathbb{R}_+$ and $\phi(\omega) = \mathsf{Var}_\omega(x)$ we take

$$\begin{aligned}\phi(\omega_{\boldsymbol{p}(a(j,k))}) &= \mathsf{Var}_\omega(x_{\boldsymbol{p}(a(j,k))}) \\ &= \mathsf{Cov}_\omega(\widetilde{x}_j, \widetilde{x}_k) = \Phi_{j,k}.\end{aligned} \tag{4}$$

The middle equality holds since, by the independence assumption, the only non-zero contribution to $\mathsf{Cov}_\omega(\widetilde{x}_j, \widetilde{x}_k)$ is due to delays on the common portion of the paths to j and k. By the independence assumption ϕ is additive and so $\phi_k = \mathsf{Var}_\omega(x_k)$, the delay variance of link k. ϕ is increasing provided delays are not constant. $\Phi_{j,k}^{(n)}$ is the sample covariance, *i.e.*, $\Phi_{j,k}^{(n)} = \frac{1}{n}\left(\sum_{i=1}^n \widetilde{x}_j^{(i)}\widetilde{x}_k^{(i)} - \frac{1}{n}\sum_{i=1}^n \widetilde{x}_j^{(i)} \cdot \sum_{i=1}^n \widetilde{x}_k^{(i)}\right)$.

3.3 Delay Distribution-Based Inference

With inference supplying the full distribution of the cumulative delay from the root to a given node, there are several choices of the increasing function ϕ available: the complementary cumulative distribution function (ccdf), the delay moments, and the delay variance.

In the formalism, $\mathcal{X} = \{q, 2q, \ldots, dq, \infty\}$, $q > 0, d \in \mathbb{N}$, with x_k the delay on link k, discretized in bins of width q. dq is a threshold delay above which packets are considered lost, x_k taking the value ∞. Composition adds delays along a path: $x_1 \otimes x_2 = x_1 + x_2$. The identity is $\boldsymbol{z} = 0$. Aggregation takes minimum delay between paths $x_1 \oplus x_2 = x_1 \wedge x_2$; hence $\widetilde{x}_k = y$ if the minimum delay from the source to some receiver descended from k is y. In [5] it was shown how a generalization of the approach for loss inference in Section 3.1 above can be used to express the discretized distribution $\omega_{\boldsymbol{p}(a(j,k))}$ of the delay from the root to an interior node, in terms of the distribution $\widetilde{\omega}_{j,k}$ aggregate delays to leaf nodes descended through offspring j, k. More precisely, denote $A_k(i) = \mathsf{P}[x_{\boldsymbol{p}(k)} = iq]$, $\gamma_k(i) = \mathsf{P}[\widetilde{x}_{\boldsymbol{p}(k)} \le iq]$, and $\gamma_{j,k}(i) = \mathsf{P}[\widetilde{x}_{\boldsymbol{p}(j)} \oplus \widetilde{x}_{\boldsymbol{p}(k)} \le iq]$, $i \in \mathcal{X}$. Then:

$$A_{a(j,k)}(0) = \frac{\gamma_j(0)\gamma_k(0)}{\gamma_j(0) + \gamma_k(0) - \gamma_{j,k}(0)} \tag{5}$$

and $A_{a(j,k)}(i)$, $i = 1, \ldots, d$, is recursively computed as the smallest solution of the following quadratic equation:

$$\begin{aligned}&\gamma_{j,k}(i) - A_{a(j,k)}(0) + A_{a(j,k)}(0) \\ &\cdot \textstyle\prod_{\ell\in\{j,k\}}\left[1 - \frac{\gamma_\ell(i) - \sum_{m=1}^{i-1}\beta_\ell(i-m)A_{a(j,k)}(m) - \beta_\ell(0)A_{a(j,k)}(i)}{A_{a(j,k)}(0)}\right] \\ &+ \textstyle\sum_{m=1}^{i-1} A_{a(j,k)}(m)\left\{\prod_{\ell\in\{j,k\}}[1-\beta_\ell(i-m)] - 1\right\} \\ &+ A_{a(j,k)}(i)\left\{\textstyle\prod_{\ell\in\{j,k\}}[1-\beta_\ell(0)] - 1\right\} = 0\end{aligned} \tag{6}$$

where $\beta_\ell(i) = \frac{\gamma_\ell(i) - \sum_{m=1}^{i} A_{a(j,k)}(m)\beta_\ell(i-m)}{A_{a(j,k)}(0)}$, $\ell \in \{j,k\}$.

In the first instance we can take $\mathcal{Q}$ as the set of ccdf's arising from the delay distributions. Excluding from Ω trivial distributions in which all link delays are zero, then since link delays are non negative, the map ϕ taking distributions to ccdf's is increasing in the sense of Definition 1(i).

In order to avoid comparing entire distributions, we can instead compare summary statistics. Since link delays are non-negative then any function of the form $\phi(\omega) = \mathsf{E}_\omega[h(x) \mid x < \infty]$ is estimable and increasing when h is an increasing function, e.g., $h(x) = x^p, p > 0$. (Here x represents a generic mark with distribution ω.) A special case is the **delay average** estimator, obtained when $p = 1$. This is additive since the mean of the sum of two random variables is the sum of their means. Another estimator is the **delay variance** estimator: $\phi(\omega) = \mathsf{Var}_\omega[x|x < \infty]$. This is additive due to the independence of link delays.

For the delay average and variance classifiers, use $\Phi_{j,k} = \mathsf{E}_\omega[x_{\boldsymbol{p}(a(j,k))} \mid x_{\boldsymbol{p}(a(j,k))} < \infty] := \sum_{i=0}^d iqA_{a(j,k)}(i) / \sum_{i=0}^d A_{a(j,k)}(i)$ and $\Phi_{j,k} = \mathsf{Var}_\omega[x_{\boldsymbol{p}(a(j,k))} \mid x_{\boldsymbol{p}(a(j,k))} < \infty] = \sum_{i=0}^d (iq)^2 A_{a(j,k)}(i) / \sum_{i=0}^d A_{a(j,k)}(i) - \mathsf{E}_\omega^2[x_{\boldsymbol{p}a(j,k)} \mid x_{\boldsymbol{p}(a(j,k))} < \infty]$, respectively, where we condition on the delay being finite. The corresponding $\Phi_{j,k}^{(n)}$ are computed using the estimated distribution $A_{a(j,k)}^{(n)}(i)$, computed through (5) and (6) using the estimates $\gamma_\ell(i)^{(n)} := n^{-1}\sum_{m=1}^n \mathbf{1}_{\{x_{\boldsymbol{p}(\ell)}^{(m)} \le iq\}}$ and $\gamma_{j,k}(i)^{(n)} := n^{-1}\sum_{m=1}^n \mathbf{1}_{\{x_{\boldsymbol{p}(j)}^{(m)} \oplus x_{\boldsymbol{p}(k)}^{(m)} \le iq\}}$ in place of $\gamma_\ell(i)$, $\ell \in \{j,k\}$, and $\gamma_{j,k}(i)$, $i \in \mathcal{X}$.

3.4 Utilization-Based Inference

In this case the increasing function ϕ is (a function of) the probability of encountering minimal delay at all links in a path. This case can be regarded as a degenerate case of the delay distribution inference in which $\mathcal{X} = \{0, 1\}$ where $x_k = 0$ indicates no delay on link k, while $x_k = 1$ corresponds to any non-zero queueing delay. Hence the fraction of packets that experience $x_k = 1$ is a direct measure of the utilization of link k. Since $x_{\boldsymbol{p}(k)} = 0$ iff $x_j = 0$ for all j in the path $\boldsymbol{p}(k)$, the setup maps exactly onto the loss inference described in Section 3.1, except with the roles of 0 and 1 interchanged.

4 Misclassification Analysis

In this section, we analyze the probabilities of misclassification for the various instances of BT, and estimate their convergence rates.

Denote by E_i the event that BT has correctly reconstructed the subtree rooted at node i, $i \in V$. Since the algorithm proceeds iteratively up the tree, E_i requires first that both the subtrees rooted at its child nodes have been correctly reconstructed, then that its child nodes have been then paired together. Therefore, for $i \in V \setminus R$ we can write

$$E_i \supseteq E_{h(i)} \cap E_{h^*(i)} \cap \left(\cap_{(l,j,k)\in\mathcal{S}(i)} Q(l,j,k)\right) \quad (7)$$

where $S(i) = \{(h(i), h^*(i), k), (h^*(i), h(i), k) | i, k \neq a(i,k)\}$ and $Q(l,j,k)$ is the event that

$$D_i^{(n)}(l,j,k) := \Phi_{l,j}^{(n)} - \Phi_{l,k}^{(n)} > 0 \quad (8)$$

holds. In $\cap_{(l,j,k)\in\mathcal{S}(i)} Q(l,j,k)$, $h(i)$ and $h^*(i)$ are grouped together to form node i for all possible ways to reconstruct the tree. Denote by E the event that the tree is correctly classified. ¿From (7) we immediately have that $E \supseteq \cap_{i\in V\setminus R} \cap_{(l,j,k)\in\mathcal{S}(i)} Q(l,j,k)$. This provides the following upper bound for the misclassification probability, denoted by

$$P^f := \mathsf{P}[E^c] \leq \sum_{i\in V\setminus R} \sum_{(l,j,k)\in\mathcal{S}(i)} \mathsf{P}[Q_i^c(l,j,k)] \quad (9)$$

Normal Approximations It can be shown that $\sqrt{n}(\Phi_{i,j}^{(n)} - \Phi_{i,j})$ has asymptotically Gaussian distribution as the number of probes $n \to \infty$ in all instances described in Section 3. See [2, 4] for the loss and delay covariance case, for the other estimators it follows from Theorem 3 in [5].

Theorem 3 *Under the conditions of Theorem 1, for each $i \in V \setminus R$, $\sqrt{n} \cdot (D_i^{(n)}(j,l,k) - D_i(j,l,k))$, $(j,l,k) \in \mathcal{S}(i)$, where $D_i(j,l,k) = \Phi_{j,l} - \Phi_{j,k}$, converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2_{D_i}(j,l,k) = \lim_{n\to\infty} n \cdot \left(\mathsf{Var}(\Phi_{j,l}^{(n)}) + \mathsf{Var}(\Phi_{j,k}^{(n)}) - 2\mathsf{Cov}(\Phi_{j,l}^{(n)}, \Phi_{j,k}^{(n)})\right)$.*

Theorem 3 suggests we can approximate $\mathsf{P}[Q_i^c(j,l,k)]$ by $\Psi\left(-\sqrt{n} \cdot \frac{D(j,l,k)}{\sigma_{D_i}(j,l,k)}\right)$, where Ψ is the cdf of the standard normal distribution. For large n, we have the following leading exponential approximation

$$\mathsf{P}[Q_i^c(j,l,k)] \approx e^{-(n/2)D_i^2(j,l,k)\sigma^2_{D_i}(j,l,k)} \quad (10)$$

where the exponent is given by the dominant term of the ratio. Since the largest term over $\cup_{i\in V\setminus R}\mathcal{S}(i)$ in (9) should dominate for large n, we expect the curve $\log P^f$ vs. n to be asymptotically linear with negative slope

$$\inf_{i\in V\setminus R} \inf_{(j,l,k)\in\mathcal{S}(i)} \frac{D_i^2(j,l,k)}{\sigma^2_{D_i}(j,l,k)} \quad (11)$$

4.1 Misclassification Probability for the Different Classifiers

The foregoing analysis can be instantiated with the different estimators to obtain the misclassification probability of the topology classifiers. In the following we compute the asymptotic behavior of the different classifiers by substituting the proper expressions in (11). In general, the calculation of the infimum in (11) is quite difficult since $\sigma^2_{D_i}(j,l,k)$ is a complex function of both the topology and the distribution ω. Here, we capture the dominant modes of misclassification in asymptotic regime of small loss and delay. The results in [2] and Theorem 3 in [5] suggest that in this regime, the curve $\log P^f$ vs. n is asymptotically linear with negative slope

1. for the loss based classifier

$$\frac{n}{2} \inf_{i\in V\setminus R} \mathsf{P}[x_i = 0] \quad (12)$$

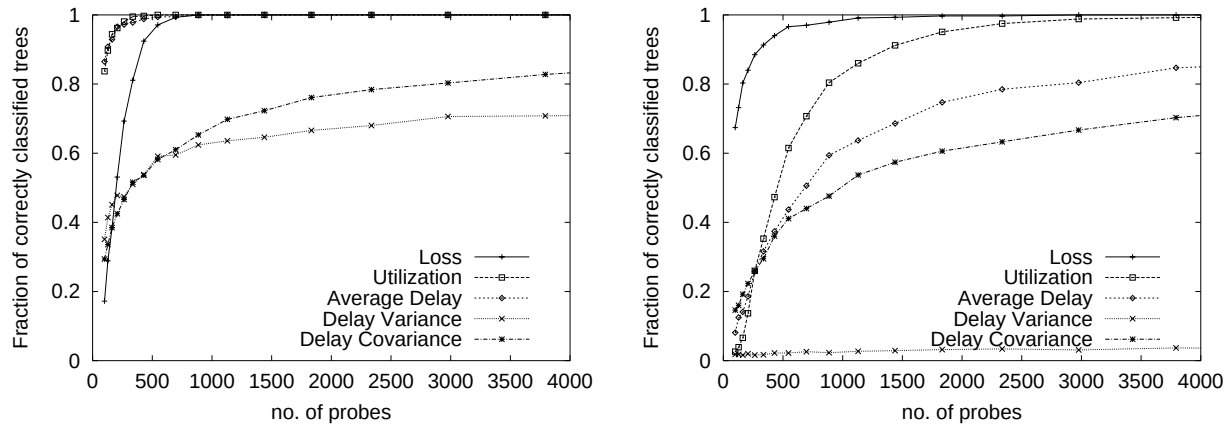


Figure 2: MODEL SIMULATION. Fraction of correctly classified topologies for different classifiers as function of the number of probes: (a) light load scenario; (b) heavy load scenario.

2. for the utilization based classifier

$$\frac{n}{2} \inf_{i \in V \setminus R} \mathsf{P}[x_i > 0] \tag{13}$$

3. for the average based classifier

$$\frac{n}{2} \inf_{\substack{i \in V \setminus R \\ k \in V \mid i, k \neq a(i,k)}} \frac{\left(\sum_{i \preceq j \prec a(i,k)} \mathsf{E}(x_i) \right)^2}{\sum_{i \preceq j \prec a(i,k)} \mathsf{E}(x_i^2)} \tag{14}$$

4. for the variance based classifier

$$\frac{n}{2} \inf_{\substack{i \in V \setminus R \\ k \in V \mid i, k \neq a(i,k)}} \frac{\left(\sum_{i \preceq j \prec a(i,k)} \mathsf{Var}(x_i) \right)^2}{\sum_{i \preceq j \prec a(i,k)} \mathsf{E}(x_i^4)} \tag{15}$$

We were not able to establish an analogous result for the covariance based approach. In this case we used our experience from experiments to determine which event dominates misclassification. We observe in most experiments that, as n increases, the most likely way to misclassify a tree is by incorrectly identifying the link with the smallest link variance; this happens by mistakenly grouping one of its child nodes with its sibling node. This suggests the following approximation for the covariance approach

$$P^f \approx e^{-(n/2) \frac{\mathsf{Var}^2(x_j)}{\sigma^2_{D_j}(h(i), h^*(i), j^*)}}, \tag{16}$$

where $j = \arg\min_{i \in V \setminus R} \mathsf{Var}(x_j)$.

5 Simulation Evaluation and Algorithm Comparison

In this section we compare the performance of the different classification algorithms through two types of simulation. In *model simulations* delay and loss are chosen to follow our statistical model, allowing us to test algorithm performance in the setting on which our analysis is based. *Network simulations*, using the `ns` [7] simulator, test the algorithms in a more realistic setting, where delay and loss are due to queueing delay and buffer overflows at nodes as multicast probes compete with background TCP/UDP traffic.

5.1 Model Simulation

In the model simulations, at each link a probe is either lost, or encounters no delay, or suffers an exponentially distributed delay. We conducted 1000 simulations over random generated 15 node binary trees. In Figure 2 we plot the fraction of correctly classified topologies as a function of the number of probes for the different classifiers. We considered two regimes: a light load regime with low loss (1%) and utilization (randomly chosen between 10% and 40%), and a heavy load regime with higher loss (randomly chosen between 1% and 20%) and utilization (randomly chosen in between 30% and 80%). In both cases, mean delays were randomly chosen between 0.2ms

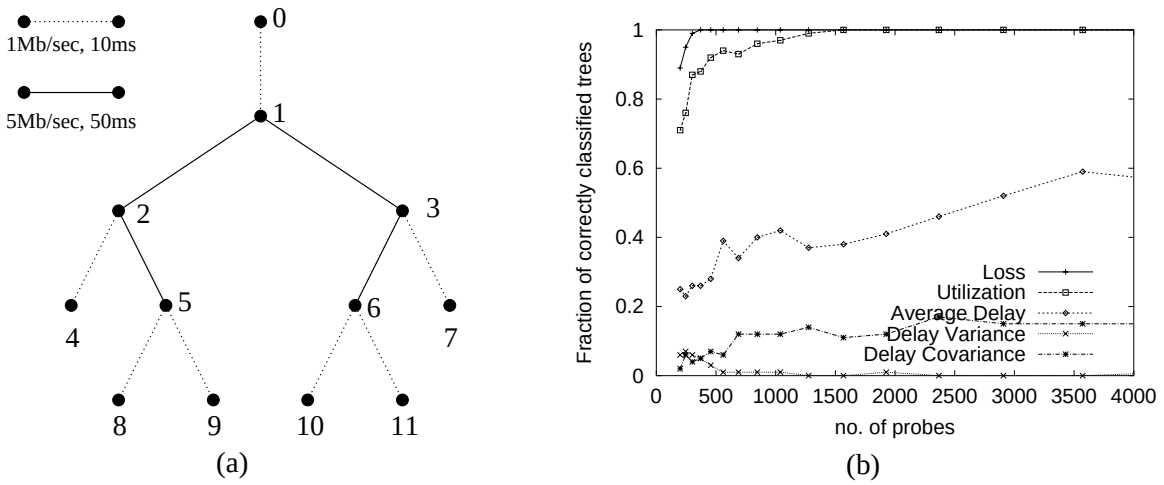


Figure 3: ns SIMULATION: (a) simulation topology; (b) fraction of correctly classified topologies for different classifiers as function of the number of probes

and 2ms. We adopted a delay granularity of 1ms. (Although this seems large compared with mean delays, the delay predictions are quite accurate; see [5]).

The loss based classifier is found to be most accurate in general. The exception is with small numbers of probes at small loss rates. Then rare losses provide insufficient data points, and accuracy is greater for the utilization and average delay classifiers. The utilization based classifier has the best accuracy among the delay distributions based classifiers. This was expected because the delay average and variance approaches use estimates of the entire delay distribution while the utilization approach uses estimates of only the first bin; estimation of the weights of lower bins being found to be more accurate. Similarly, delay average is more accurate that delay variance since it attaches less weight to higher delay bins.

From the plots, the utilization based approach appears to work very well with low links utilizations, while its performance degrades with higher utilizations, which is in contrast with (13). This can be explained by observing that the theorem holds for the limit behavior as utilization goes to 0; in our experiments, we found that (13) captures well the misclassification behavior for utilization up to 20%. On the other hand, as link utilizations increase, the number of events used by the algorithm, namely those of minimum end-to-end delay, decreases rapidly. This results in increased estimator variance.

The two best performing algorithm, (loss and utilization based) have the smallest computational complexity. All algorithms require $O(\#R^3)$ node pairs computations. Each of these is $O(n)$ for the loss, utilization and covariance based estimators. These are considerably more complex for the delay average and variance classifiers since the whole delay distribution must be calculated, by recursively solving quadratic equations, the number of which is inversely proportional to the bin size q.

5.2 TCP/UDP Network Simulation

The ns simulations used the topology shown in Figure 3(a). To capture the heterogeneity between edges and core of a WAN, interior links have higher capacity (5Mb/sec) and propagation delay (50ms) then at the edge (1Mb/sec and 10ms). Each link is modeled as a FIFO queue with a 4-packet capacity.

The root node 0 generates probes as a 20Kbit/s stream comprising 40 byte UDP packets according to a Poisson process with a mean interarrival time of 16ms; this represents 2% of the smallest link capacity. The background traffic comprises a mix of infinite data source TCP connections (FTP) and exponential on-off sources using UDP. Averaged over the different simulations, the link loss ranges between 1% and 11% and link utilization ranges between 20% and 60%. The average delay ranges between 1 and 2ms for the slower links and between 0.2 and 0.5ms

for the faster links. The delay distributions were computed using a bin size of 1ms.

In Figure 3(b) we plot the fraction of correctly identified topologies over 100 simulations. The relative accuracy among the different classifiers is in agreement with the results from the model simulation with the loss based algorithm having the best performance with no misclassification for more than 500 probes. The rather poor performance of the delay based algorithms, with the exception of the utilization classifiers, is largely due to the presence of spatial correlation. In our simulations, a multicast probe is more likely to experience a similar level of congestion on consecutive links or on sibling links than is dictated by the independence assumption. This has negative impact on the accuracy of the delay estimates which accounts for the observed performance.

We also observed temporal correlation among successive probes that encountered the same congestion events. However, it can be shown that the presence of short-term correlation does not affect estimator consistency, although the convergence rate may be slowed.

6 Conclusions

In this paper we have presented a general framework for the inference of the multicast tree topologies from end-to-end measurements. In contrast with tools such as `mtrace` [6], cooperation of intervening network nodes is not required.

We specified an algorithm which reconstructs the topology of multicast tree in presence of any packet performance measure that: (i) monotonically increases as the packet traverses down the tree; and (ii) that can be estimated on the basis of end-to-end measurements at the receivers. Building on previous results in [1, 4, 5], we were able to specify several instances of this algorithm based on performance measures as packet loss, link utilization, delay average and delay variance.

We investigated the statistical properties of the algorithms, and showed that, under mild assumptions, they are consistent and computed their convergence rate. We evaluated our classifiers though simulation. We found out that the two algorithms with the lowest computational complexity, namely, the loss based and the utilization based algorithm, also have the best performance, with the loss based algorithm being in general the most accurate except when the number of probes and the loss rate are both small. Moreover, both algorithms seemed to be robust and exhibit good convergence in real traffic simulations, in spite of violation of the independence assumption of our model.

Finally, the algorithms described in this paper are each based on a different performance metric. We are currently extending our work by studying algorithms which fully take advantage of the available measurements by possibly integrating the different performance metrics we have here separately considered.

References

[1] R. Caceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics", IEEE Trans. on Information Theory, November 1999.

[2] R. Caceres, N.G. Duffield, J.Horowitz F. Lo Presti and D. Towsley, "Statistical Inference of Multicast Network Topology", in Proc. 1999 IEEE Conf. on Decision and Control, Phoenix, AZ, Dec. 1999.

[3] Cooperative Association for Internet Data Analysis, "Internet Measurement Efforts," `http://www.caida.org/Tools/taxonomy.html#InternetMeasurement`

[4] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", Proceedings IEEE Infocom 2000, Tel Aviv, March 2000, to appear.

[5] F. Lo Presti, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[6] `mtrace` – Print multicast path from a source to a receiver. See `ftp://ftp.parc.xerox.com/pub/net-research/ipmulti`

[7] `ns` – Network Simulator. See `http://www-mash.cs.berkeley.edu/ns/ns.html`

[8] S. Paul, et al. "Reliable multicast transport protocol (RMTP)", *IEEE JSAC*, Vol. 15, No. 3, pp. 407–421, April 1997.

[9] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, Vol. 36, No. 8, pp. 48-54, August 1998.

[10] S. Ratnasamy & S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", Proc. IEEE Infocom'99, New York, NY (1999)

Adaptive Multicast Topology Inference

N.G. Duffield[1] J. Horowitz[2] F. Lo Presti[1,3]

[1] AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
{duffield,lopresti}@research.att.com

[2] Dept. Math. & Statistics
University of Massachusetts
Amherst, MA 01003, USA
joeh@math.umass.edu

[3] Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA

Abstract— **The use of end-to-end multicast traffic measurements has been recently proposed as a means to infer network internal characteristics as packet link loss rate and delay. In this paper, we propose an algorithm that infers the multicast tree topology based on these end-to-end measurements. Differently from previous approaches which make only partial use of the available information, this algorithm adaptively combines different performance measures to reconstruct the topology. We establish its consistency and evaluate its accuracy through simulation. We show that in general it requires many fewer probes to correctly identify the topology than other methods.**

Keywords. End-to-end measurements, Topology Discovery, Adaptive, Estimation Theory, Multicast Tree.

I. Introduction

Background and Motivation. As communications networks grows in size and complexity, it has become increasingly important to measure their performance. To overcome the limitations imposed by administrative diversity which *de facto* prevents general direct access to large portions of the network, there has been increasing interest in approaches that aim to characterize the network internal behavior from the sole external end-to-end measurements. Currently, there are several measurements infrastructure projects (including CAIDA [2], Felix [9], IPMA [10], NIMI [15] and Surveyor [18]) that collect and analyze end-to-end measurements across a mesh of paths between hosts.

In these approaches, a fundamental design issue is the type of measurements to be performed across the network and the methodology adopted to infer the internal network behavior in terms of the performance experienced by the measurements hosts. A promising approach, MINC (*Multicast Inference of Network Characteristics*), relies on the use of multicast end-to-end measurements. In contrast to unicast traffic, multicast traffic introduces a well structured correlation in the end-to-end behavior observed by the receivers that share the same multicast session. This in turn allows to draw inferences about the performance characteristics of the internal links without the cooperation of network elements in the path such as packet loss rates, [3], packet delay distributions, [11], and packet delay variance, [6]. There is ongoing work [1] to incorporate some of these techniques into the NIMI measurements infrastructure.

All these inference methods require knowledge of the multicast tree topology. Unfortunately, this is typically unknown. This motivates the need for algorithms that can identify the topology of the tree. Another motivation is that knowledge of the multicast topology can be of use to multicast applications. There are several reliable multicast protocols (e.g., RMTP[14]) which organize receivers in logical hierarchies using the underlying topology, if possible. Other applications attempt to identify receivers that share the same network bottleneck [16].

Several algorithms have been proposed for identifying multicast topologies based on the sole loss observations at receivers. An algorithm for inferring the topology of a binary tree was first proposed in [16]. The main idea was the simple observation that as the number of packets grows multicast receivers sharing a longer portion of the multicast distribution tree also have higher shared loss rates; this information could in turn be used to reconstruct the topology by recursively grouping the pair of nodes with the highest shared loss. In [8] the correctness this algorithm was proven and the approach was extended to general topologies by introducing several other loss-based algorithms. More recently, algorithms have been proposed for identifying multicast topologies based on delay measurements instead. By observing that the approach in [16] and [8] can be generalized to any performance measures that (i) monotonically increases as the packet traverse the tree, and (ii) can be estimated on the sole basis of end-to-end measurements at the receivers, in [7] several algorithms are specified based on delay performance measures as link utilization, delay average and delay variance.

The accuracy of these approaches is limited by the fact that each of the above algorithm reconstructs the topology using only the information provided by one single performance measure, *e.g.*, loss rates or delay averages, thus making only partial use of the available measurements. In addiction, as shown in [7], no algorithm appears to perform better than the others in general. Our experience has shown that typically under moderate and heavy load network conditions (high link loss and utilization) the loss based algorithm is generally the most accurate while under light load condition (low link loss and utilization), the algorithm based on link utilization performs best. Therefore, it is then not clear which algorithm could be best suited to reconstruct multicast topologies across large internetworks where different portions of the network can experience quite different conditions. In the most general case, the different algorithms could yield quite different reconstructed topologies; clearly, a method which allows to choose among them or better to compose them is much desired.

Contribution. In this paper we propose a new algorithm for identifying multicast topologies based on joint loss and delay measurements at the receivers. This algorithm combines the different performance measures and reconstruct the tree by adaptively choosing step by step that which insures the best accuracy. Intuitively, by so doing we compose the topologies each performance measure would yield by choosing for each portion of the tree its more accurate reconstruction.

* This work was supported by in part by DARPA and the AFL under agreement F30602-98-2-0238

The key contribution underlying this approach is the ability to determine which performance measure minimizes the probability of making an error. We propose a technique for estimating the probability of incorrect identification of the topology. This is accomplished by a careful enumeration of all the possible erroneous decisions and by estimating the probability of each of them. We also analyze the modes of misclassification and verify that our estimate converges to the true error probability as the number of packets increases. Therefore we can use this estimate to determine the level of accuracy of a given reconstructed topology, or more importantly, the number of probe packets required to achieve a desired level of accuracy.

We establish that the joint algorithm is consistent, *i.e.* the probability of correctly identifying the topology converges to 1 as the number of probes grows to infinity. Analysis of a simple scenario shows that the joint algorithm can significantly outperform any of the algorithms previously considered. We also use simulation to evaluate its accuracy. In all the scenarios considered, we find that the joint algorithm has the best performance, requiring in general many fewer probes to correctly identify the topology than other methods.

In this paper, we will restrict our attention to topology inference based solely on loss and utilization performance measures. A first reason is simplicity; as later shown, the loss process and the utilization process are formally identical once we substitute the event of "packet not lost" with the event of "packet not delayed"; as a consequence the very same results apply in both cases. A second reason is that they also have the lowest computational complexity. Finally, they are the most accurate: as previously mentioned, in most cases, either the loss based or the utilization based algorithms has the best performance. Hence, while the joint algorithm extends to accommodate other performance measures, in practice most of the benefit is achieved by combining the loss and utilization estimators.

Implementation Requirement. In contrast to loss, delay measurements require the deployment of measurements hosts with synchronized clocks. Global Positioning System (GPS) which is used in some of the mentioned measurements infrastructures allows accuracy within tens of microseconds. This is sufficient for accurate utilization measurements which, in particular, require the accurate assessment of the minimum end-to-end delay. We believe this is not the case for the more widely deployed Network Time Protocol [12], which only provides accuracy on the order of tens of milliseconds.

Structure of the Paper. The rest of the paper is organized as follows. In Section II and III we review our model and the loss and utilization topology inference algorithms In Section IV we introduce the joint loss/utilization algorithm; we also describe the technique for estimating the probability of topology misclassification. In Section V we analyze the performance of the different algorithms. Their accuracy is then evaluated in Section VI through simulation. We conclude in Section VII; some proofs are deferred to the Appendix.

II. Model & Inference

Tree Model. The physical multicast tree comprises actual network elements (the nodes), and the communication links than join them. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except the leaf nodes and possibly the root, must have 2 or more children. We can construct the logical tree from the physical tree by deleting all links with one child (except for the root) and adjusting the links accordingly by directly joining its parent and child.

Let $\mathcal{T} = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify the root node 0 with the source of probes, and $R \subset V$ will denote the set of leaf nodes (identified as the set of receivers). The set of children of node $k \in V$ is denoted by $d(k)$. For each node k, other than the root 0, there is a unique node $f(k)$, the *parent* of k, such that $(f(k), k) \in L$. We will refer to the link $(f(k), k)$ as simply link k. We shall define $f^n(k)$ recursively by $f^n(k) = f(f^{n-1}(k))$ with $f^1 = f$. We say that j is a descendant of k if $k = f^n(j)$ for some integer $n > 0$, and write the corresponding partial order in V as $j \prec k$. $a(i, j)$ will denote the minimal common ancestor of i and j in the $\preceq$-ordering. For $k \in V$ we let $\mathcal{T}(k) = (V(k), L(k))$ denote the subtree of $\mathcal{T}$ that is rooted at k, and set $R(k) = R \cap V(k)$.

Delay and Loss Model. Probe packets are dispatched down the tree from the root node 0. With multicast, each probe arriving at anode k gives rises to copy sent to each child node of k. On each link, the packet is either lost, or transmitted with some delay. We regard the delay as the sum of two components: a fixed propagation delay, and a variable queueing delay. We represent the latter by a random variable $Z_k \in [0, \infty]$ that specifies the queueing delay encountered by a packet attempting to traverse link k, with $Z_k = \infty$ signifying packet loss. By convention $Z_0 = 0$. The accrued queueing delay for the path from the root to a node k is $Y_k = \sum_{j \succeq k} Z_k$. This yields the property that $Y_k = \infty$ for a packet lost on some link between node 0 and k; likewise $Y_k = 0$ if no queueing delay is encountered on any link of the path.

Let $\alpha_l(k) = \mathsf{P}[Z_k < \infty]$ denote the probability of transmission on link k, and $\alpha_u(k) = \mathsf{P}[Z_k = 0]$ the probability of transmission with no queueing delay. A tree is said to be *canonical* if for all links k, $0 < \alpha_u(k) \leq \alpha_l(k) < 1$. A tree can be reduced to canonical form by (i) removing each link k for which with $\alpha_l(k) = 1$ or $\alpha_u(k) = 1$ and identifying its endpoints; and (ii) pruning all subtree descended from links that have $\alpha_l(k) = 0$ or $\alpha_u(k) = 0$. Henceforth we work exclusively with canonical trees; only for these are the link characteristics uniquely identifiable.

Loss and Utilization Processes. Here it suffices to analyze a projection of the delay processes Z_k. For each $k \in V$ let $X_l(k) = \mathbf{1}_{\{Y(k)<\infty\}}$. We call $X_l = (X_l(k))_{k \in V}$ the **loss process**: $X_l(k) = 1$ if the probe reaches k and 0 otherwise. For each $k \in V$ let $X_u(k) = \mathbf{1}_{\{Y(k)=0\}}$. We call $X_u = (X_u(k))_{k \in V}$ the **utilization process**: $X_u(k) = 1$ if the probe reaches k with no queueing delay, and 0 otherwise. The name arises since link queueing delay is zero iff the link is not utilized: $1 - \alpha_u(k)$ is hence the link utilization.

We assume the Z_k are independent random variables. Then X_u and X_l are Markov processes on $\mathcal{T}$. Their structure is for-

mally identical. The loss process satisfies

$$X_l(0) = 1; \qquad X_l(f(k)) = 0 \Rightarrow X_l(k) = 0;$$
$$\mathsf{P}[X_l(k) = 1 \mid X_l(f(k)) = 1] = \alpha_l(k). \tag{1}$$

The utilization process is formally identical upon replacing the event of "no loss" with that of "no delay". Then (1) holds when X_l, α_l are replaced by X_u, α_u. In the rest of the paper we will omit the subscripts l and u when the same statement holds for both cases.

Inference of Shared Path Characteristics. When probes are sent down the tree we cannot observe the entire processes X but only the outcomes at the receivers $(X(k))_{k\in R}$. By exploiting the correlation of multicast traffic, in [3] it was shown how the link loss rates can be computed from the distribution of $(X(k))_{k\in R}$ when the topology is known. Here, to infer the topology, we will use the following generalization of the results in [3].

Let $A(k) = \prod_{j\succeq k} \alpha(j)$ denote the probability that a probe reaches node k (the $A_l(k)$ version) or reaches is without queueing delay (the $A_u(k)$ version). A short probabilistic argument shows that for any two nodes i and $j, i, j \neq a(i,j)$,

$$A(k) = A(i,j) := \frac{\mathsf{P}[\vee_{\ell\in R(i)} X(\ell) = 1]\mathsf{P}[\vee_{\ell\in R(j)} X(\ell) = 1]}{\mathsf{P}[\vee_{\ell\in R(i)} X(\ell) = \vee_{\ell\in R(j)} X(\ell) = 1]} \tag{2}$$

where $k = a(i,j)$. (2) expresses the behavior along the shared portion of the path from the source to a pair of nodes in terms of the probabilities of leaf-measurable events.

To infer the probabilities from measurements, consider an experiment in which a set of n probes is dispatched from the source. From the outcomes $(x^{(1)}, \ldots, x^{(n)})$ with $x^{(m)} = (X^{(m)}(k))_{k\in R}$, we can estimate $A(k)$ by substituting the probabilities in (2) by their empirical means, obtaining

$$A^{(n)}(i,j) = \frac{\sum_{m=1}^n X^{(m)}(i) \cdot \sum_{m=1}^n X^{(m)}(j)}{n \cdot \sum_{m=1}^n X^{(m)}(i) \cdot X^{(m)}(j)} \tag{3}$$

where we define $X^{(m)}(k) := \vee_{\ell\in R(k)} X^{(m)}(\ell)$. It is possible to show that $A^{(n)} = (A^{(n)}(i,j))_{i,j\in V}$ is consistent ($A^{(n)} \overset{n\to\infty}{\longrightarrow} A$ with probability 1) and, as n goes to infinity, $\sqrt{n}(A^{(n)} - A)$ converges in distribution to a multivariate Gaussian random variable with mean 0 and covariance matrix $\sigma_A = \sigma_A(A)$. Details can be found in [8].

A complication arises in case of utilization estimation as we have to account for (i) the presence of the fixed delay component in the experimental data due to propagation delays and (ii) the inherent limitation of time measurements accuracy due to clocks resolution. To this end, we (i) normalize each measurement by subtracting the minimum delay seen at the leaf and (ii) introduce a tolerance τ (typically smaller than $1ms$) in deciding whether a given delay is a "minimum" delay. In other words, operationally we define $X_u^{(m)}(k) = \mathbf{1}_{\{Y^{(m)}(k) - \min_{l=1}^n Y^{(l)}(k) \le \tau\}}$ where $Y^{(m)}(k)$ is the delay experienced by the m^{th} probe sent to receiver k. This amounts to assign the observed minimum delay as the propagation delay, under the assumption that at least one probe has experienced no queueing delay along the path.

1. *Input*: The set of receivers $R = \{i_1, \ldots, i_r\}$
2. $R' := R$; $V' := R'$; $L' = \emptyset$;
3. **while** $|R'| > 1$ **do**
4. $\quad U :=$ **select pair** ;
5. $\quad V' := V' \cup \{U\}$;
6. $\quad L' := L' \cup \{(U, \ell) : \ell \in U\}$;
7 $\quad \alpha(\ell) = A(\ell)/A(j,k)$, $\ell \in U$;
8. $\quad R' := (R' \setminus U) \cup \{U\}$;
9. **enddo**
10. $V' := V' \cup \{0\}$; $L' = L' \cup \{(0, R')\}$;
11. *Output*: tree (V', L') ;
12. **procedure select pair**
13. $\quad$ **return** $U = \{j,k\} \subseteq R'$ with minimal $A(j,k)$;
14. **end procedure**

Fig. 1. Deterministic Binary Tree Classification Algorithm (DBT).

III. Loss and Utilization Topology Inference

Deterministic Reconstruction of Binary Trees. Our approach to loss (or utilization) topology inference relies on being able through (2) to identify the characteristics along internal paths of the multicast tree from the probability of measurable events at receivers. The key observation is that $a(j,k) \prec a(j',k')$ implies $A(j,k) < A(j',k')$, from which it follows that the pair $\{j,k\} \subset R$ which has minimal $A(j,k)$ is a sibling pair; a short argument shows that if not, $A(j,k)$ would not be minimal. The idea is to proceed recursively, starting from the receivers, by adding the parent node as sibling are identified. This approach is formalized in the Deterministic Binary Tree Classification Algorithm (DBT); see Figure 1.

DBT operates as follows. R' denotes the current set of nodes from which a pair of siblings will be chosen, initially equal to the receiver set R. We first use the procedure *select pair* below

procedure select pair
$\quad$ **return** $U = \{j,k\} \subseteq R'$ with minimal $A(j,k)$;
end procedure

to find the pair $U = \{j,k\}$ that minimizes $A(j,k)$ (line 4). This identifies the members of U as siblings, and the set U is used to represent their parent. Correspondingly, we add U to the list V' of nodes (line 5), $(U,j), (U,k)$ to the list L' of links (line 6), compute $\alpha(j)$ and $\alpha(k)$ by taking the appropriate quotient (line 7) and replace j and k by U in the set R' of nodes available for pairing in the next stage (line 8). This process is repeated until all sibling pairs have been identified (loop from line 3). Finally, we adjoin the root node 0 and the link joining it to its single child (line 10).

We say that DBT reconstructs the binary logical multicast tree (V, L) if given the receiver set R it produces (V, L) as its output.

Theorem 1: Let $\mathcal{T}$ be a binary tree. Then DBT reconstructs $\mathcal{T}$.

We postpone the proof to the Appendix.

Reconstruction of Binary Trees from Measurements. It is straightforward to derive from DBT an algorithm that estimates the topology from the end to end measurements $(x^{(1)}, \ldots, x^{(n)})$. The idea is to estimate $\mathcal{T}$ by the topology $\mathcal{T}^{(n)}$

obtained by using the estimates $A^{(n)}(j,k)$ in place of $A(j,k)$. This amounts to modifying the procedure *select pair* as follows

procedure select pair
  return $U = \{j,k\} \subseteq R'$ with minimal $A^{(n)}(j,k)$;
end procedure

Computation of $A^{(n)}(j,k)$ is accomplished via (3); to this end, observe that $X^{(m)}(k) = \vee_{\ell \in d(k)} X^{(m)}(\ell)$, so they can be recursively computed as the tree is reconstructed. It therefore suffices to add the line

4a. **foreach** $m = 1, \ldots, n$ **do** $X^{(m)}(U) = X^{(m)}(j) \vee X^{(m)}(k)$;

We call the resulting algorithm the Binary Tree Classification Algorithm (BT).

Theorem 2: With probability 1, $\mathcal{T}^{(n)} = \mathcal{T}$ for sufficiently large n. Hence $\mathcal{T}^{(n)}$ is a consistent estimator of $\mathcal{T}$, *i.e.*, $\lim_{n\to\infty} \mathsf{P}[\mathcal{T}^{(n)} \neq \mathcal{T}] = 0$.

Proof of Theorem 2: Since $A^{(n)}(j,k)$ converges almost almost surely to $A(j,k)$, then, with probability 1, for all sufficiently large n, the relative ordering of the $A^{(n)}(j,k)$ is the same as that of the $A(j,k)$ for pairs j,k for which the $A(j,k)$ are distinct. Hence, for all n sufficiently large, BT reconstructs the tree in the same manner as DBT, except possibly varying the order in which it groups pairs $\{j,k\}$ with identical $A(j,k)$. The last two statements then directly follow by standard results. ∎

Finally, observe that in line 7 BT computes an estimate $\alpha^{(n)}(\ell) = A^{(n)}(U)/\alpha^{(n)}(\ell)$ of $\alpha(\ell)$. From Theorem 2 then it immediately follows that as n goes to infinity $\alpha^{(n)}(\ell)$ converges with probability 1 to $\alpha(\ell)$.

Extension to General Trees. Inference of general trees can be accomplished by extending BT. In [8] we propose and analyze different alternatives. The simplest approach, which also turns out to be the most computationally efficient and accurate, proceeds in two steps: first it reconstructs a binary tree using BT; then it applies a threshold ε and prune all links k such that $\alpha^{(n)}(k) > 1 - \varepsilon$. The idea comes from the observation that the application of DBT to an arbitrary tree results in a binary tree in which all links k which do not exists in the original tree satisfy $\alpha(k) = 1$. In BT, the use of a threshold ε accounts for the statistical variability of the estimates.

IV. A Joint Loss-Utilization Algorithm

We now extend the framework for topology inference by proposing an algorithm which combines loss and utilization measurements. We contrast this to BT which is based on a single performance measure. The idea consists in reconstructing the topology by adaptively choosing at each step the performance measures which insures the best accuracy. We describe the algorithm below. The algorithm bases its decisions on estimates of the probability of misclassification. In the remainder of the section we will present a technique for estimating this probability.

The Joint Loss-Utilization Classification Algorithm. The joint algorithm proceeds like BT by recursively grouping nodes starting from the set of receivers. Differently from BT, here we choose at each step the performance measure on which to base the grouping decision; more precisely, at each step we determine the two pairs that minimize ${A^{(n)}}_l(.,.)$ and ${A^{(n)}}_u(.,.)$ and group that which also minimizes the probability of making an error. Specifically, we modify the procedure *select pair* as follows

procedure select pair
  foreach $X \in \{l,u\}$
    select $U_X = \{j_X, k_X\} \subseteq R'$ with minimal ${A^{(n)}}_X(j_X, k_X)$;
  return $U = \{j,k\} = \text{argmin}_{\{j_X,k_X\}, X\in\{l,u\}} P^{f,(n)}_{X,R'}$;
end procedure

where $P^{f,(n)}_{X,R'}$ denotes the (estimated) probability of misclassification, given the current set of nodes R', pairing nodes according to performance measure X. We will detail how to compute this estimate in Section IV-A.

We call the resulting algorithm the Joint Binary Tree Classification Algorithm (JBT). Denote $\mathcal{T}_j^{(n)}$ the topology obtained by JBT.

Theorem 3: With probability 1, $\mathcal{T}_j^{(n)} = \mathcal{T}$ for sufficiently large n. Hence $\mathcal{T}_j^{(n)}$ is a consistent estimator of $\mathcal{T}$, *i.e.*, $\lim_{n\to\infty} \mathsf{P}[\mathcal{T}_j^{(n)} \neq \mathcal{T}] = 0$.

We formalize the proof in the Appendix. The intuition beyond the proof is that, for all sufficiently large n, with probability 1, the relative ordering of the $A^{(n)}(j,k)$ is the same as that of $A(j,k)$ (which observe can be different for loss and utilization) from which it follows that the two pairs of nodes which minimize $A_l(.,.)$ and $A_u(.,.)$ are both siblings pairs.

Extension to General Trees. Inference of general trees is accomplished by reconstructing a binary tree using JBT first and by then pruning all links k such that $\alpha_l^{(n)}(k) > 1 - \varepsilon_l$ and $\alpha_u^{(n)}(k) > 1 - \varepsilon_u$, where we use possibly different loss and utilization thresholds, ε_l and ε_u. The estimates $\alpha_l^{(n)}(k)$ and $\alpha_u^{(n)}(k)$ are computed in line 7 of JBT by taking the appropriate ratio.

A. Estimation of the Misclassification Probability

In this section we describe the estimate of the probability of misclassification that is used in JBT. Classification proceeds by a sequence of comparison operations; the analysis of misclassification is therefore potentially complex due to the need to analyze a large number of statistically dependent modes of failure. Our approach to this is to divide and conquer. Correct classification requires correct ordering of quantities $A(j,k)$ in a number of comparison. For each such comparison, we approximate the probability of incorrect ordering in terms of the tail probability of a Gaussian random variable whose variance we calculate. For large numbers of probes, the probability of misclassification is dominated by the largest such misordering probability.

The generic comparison involves three nodes j, k and l, where $a(j,k) \neq a(j,l)$. Since $a(j,k) \prec a(j,l)$ iff $A(j,k) < A(j,l)$, the correct descendency relation between $a(j,k)$ and $a(j,l)$ is identified if

$$D^{(n)}(j,k,l) := A^{(n)}(j,l) - A^{(n)}(j,k) \tag{4}$$

has the same sign as its deterministic counterpart $D(j,k,l) = A(j,l) - A(j,k)$. Let $Q(j,k,l)$ denote this event.

The following theorem, essentially proved for loss-based classification in [8], characterizes the asymptotic behavior of $D^{(n)}(j,k,l)$ first for large n, then for small loss and delays. Denote $\overline{\alpha} = 1 - \alpha$ and let $s(k) := \sum_{l \preceq k} \overline{\alpha}(k)$.

Theorem 4: For each triple (j,k,l), $\sqrt{n} \cdot (D^{(n)}(j,k,l) - D(j,k,l))$, (j,k,l), converges in distribution, as the number of probes $n \to \infty$, to a Gaussian random variable with mean 0 and variance $\sigma^2(j,k,l)$. Moreover, as $\|\overline{\alpha}\| = \max_{k \in V} \overline{\alpha}(k) \to 0$:
(i) $D(j,k,l) = s(a(j,l)) - s(a(j,k)) + O(\|\overline{\alpha}\|^2)$;
(ii) $\sigma^2(j,k,l) = |s(a(j,l)) - s(a(j,k))| + O(\|\overline{\alpha}\|^2)$;

Measurements yield the statistic $D^{(n)}(j,k,l)$ with which to infer the descendency relations. From this we would infer $a(j,k) \succ a(k,l)$ if and only if $D^{(n)}(j,k,l) > 0$. Misordering occurs when $D(j,k,l)$ and $D^{(n)}(j,k,l)$ have opposite signs. For large n, Theorem 4 suggests the following approximation for the probability of misordering

$$\mathsf{P}[Q^c(j,k,l)] \approx \Psi\left(-\sqrt{n}\frac{|D(j,k,l)|}{\sigma(j,k,l)}\right) \tag{5}$$

where Ψ is the cdf of the standard normal distribution. Since $D(j,k,l)$ and $\sigma^2(j,k,l)$ are unknown, we need to estimate them first. The idea is to simply estimate $D(j,k,l)$ by $D^{(n)}(j,k,l)$. For the variance, we use the fact that $\sigma^2(j,k,l)$ is a continuous function $\mathcal{D}_{jkl}$ of A, $\sigma^2(j,k,l) = \mathsf{Var}[A^{(n)}(j,l)] + \mathsf{Var}[A^{(n)}(j,k)] - 2\mathsf{Cov}[A^{(n)}(j,l), A^{(n)}(j,k)] = (\sigma_{A(j,l)(j,l)} + \sigma_{A(j,k)(j,k)} - 2\sigma_{A(j,l)(j,k)})/n = \mathcal{D}_{jkl}(A)$, and estimate it by $\sigma^{(n)2}(j,k,l) = \mathcal{D}_{jkl}(A^{(n)})$. We thus approximate the probability of incorrect ordering $P[Q^c(j,k,l)]$ by

$$P^{f,(n)}_{jkl} := \Psi\left(-\sqrt{n}\frac{|D^{(n)}(j,k,l)|}{\sigma^{(n)}(j,k,l)}\right) \tag{6}$$

where we used in place of $D(j,k,l)$ and $\sigma^2(j,k,l)$ their estimates. The accuracy of (6) relies on the convergence of the estimates $D^{(n)}(j,k,l)$ and $\sigma^{(n)2}(j,k,l)$. We will verify this in Section VI.

Misclassification Probability Estimate. Consider now the ℓ-th step of JBT(or BT) and denote by $R^{(n)}_\ell$ the current set of nodes and $\{j_n, k_n\} \subset R^{(n)}_\ell$ the pair with minimal $A^{(n)}(j_n,k_n)$. This pair is chosen on the basis of the orderings $D^{(n)}(j,k,l) > 0$ for each triple $(j,k,l) \in \mathcal{S}(R^{(n)}_\ell) = \{(j_n,k_n),(k_n,j_n)\} \times (R^{(n)}_\ell \setminus \{j_n,k_n\})$. With each such ordering we associate a misordering probability $P^{f,(n)}_{jkl}$ as in (6). From the union bound $\mathsf{P}[\cup_{\mathcal{S}(R^{(n)}_\ell)} Q^c(j,k,l)] \leq \sum_{\mathcal{S}(R^{(n)}_\ell)} \mathsf{P}[Q^c(j,k,l)]$ we associate with the selection of (j_n,k_n) an estimated misclassification probability through the sum

$$\begin{aligned} P^{f,(n)}_{R^{(n)}_\ell} &= \sum_{(j,k,l)\in\mathcal{S}(R^{(n)}_\ell)} P^{f,(n)}_{jkl} && (7)\\ &\approx \max_{(j,k,l)\in\mathcal{S}(R^{(n)}_\ell)} P^{f,(n)}_{jkl} && (8)\\ &= \Psi\left(-\sqrt{n}\min_{(j,k,l)\in\mathcal{S}(R^{(n)}_\ell)} \frac{|D^{(n)}(j,k,l)|}{\sigma^{(n)}(j,k,l)}\right). && (9)\end{aligned}$$

This is the misclassification estimate we use in JBT. The approximation arises because for large n, the term with the smallest argument $|D^{(n)}(j,k,l)|/\sigma^{(n)}(j,k,l)$ will dominate the rest. Observe that $P^{(n)}_{R^{(n)}_\ell}$, $D^{(n)}(j,k,l)$ and $\sigma^{(n)}(j,k,l)$ can be directly computed from $\{A^{(n)}(j,k)\colon \{j,k\} \in R^{(n)}_\ell\}$. Furthermore, when selecting between the loss and utilization methods during step ℓ, we need only select that with the smallest composite argument $\min_{(j,k,l)\in\mathcal{S}(R^{(n)}_\ell)} |D^{(n)}(j,k,l)|/\sigma^{(n)}(j,k,l)$.

Topology Misclassification Probability Estimate. (7) associates a misclassification probability estimate with a single grouping decision. Using a simple union bound argument, we can also associate a misclassification probability estimate with the entire reconstructed topology $\mathcal{T}^{(n)}$. In JBT, since we group the pair of nodes which yields the smallest $P^{f,(n)}_{R^{(n)}_\ell}$, we can estimate the topology misclassification probability by summing over the minimum between the loss and utilization misclassification estimates,

$$P^{f,(n)}_j := \sum_{\ell=1}^{|V\setminus R|-1} \min\{P^{f,(n)}_{l,R^{(n)}_\ell}, P^{f,(n)}_{u,R^{(n)}_\ell}\}. \tag{10}$$

It is easy to realize that we can also associate a misclassification probability estimate to the topology inferred by BT. The difference is that it is simply computed by summing over (7), *i.e.*, $P^{f,(n)} := \sum_{\ell=1}^{|V\setminus R|-1} P^{f,(n)}_{R^{(n)}_\ell}$. In Section VI we will illustrate applications of these estimates.

V. Analysis of Classifier Performance

A. Performance of Single Classifier using BT

The analysis of the actual misclassification probabilities mirrors much of the previous analysis. Consider a node $i \in V$ which is to be identified during the step ℓ of BT. Let $h(i)$ and $h^*(i)$ denote its two children. Correct identification of i occurs if neither $h(i)$ nor $h^*(i)$ is incorrectly paired with some other element of R_ℓ, the set of nodes available for pairing at step ℓ. Thus, the event of correct classification at step ℓ is $Q_\ell = \cap_{(j,k,l)\in\mathcal{S}(R_\ell)} Q(j,k,l)$ where $\mathcal{S}(R_\ell) = \{(h(i),h^*(i)),(h^*(i),h(i))\} \times (R_\ell \setminus \{h(i),h^*(i)\})$. Correct classification of the whole tree is the event $Q = \cap_{\ell=1}^{|V\setminus R|-1} Q_\ell$.

Now, the various $Q(j,k,l)$ are not independent events, and neither are the Q_ℓ. However, we can use union bounds to bound above the probability of misclassification:

$$\begin{aligned} P^f &:= \mathsf{P}[Q^c] \leq \sum_{\ell=1}^{|V\setminus R|-1} P^f_{R_\ell}, \quad \text{where} && (11)\\ P^f_{R_\ell} &:= \mathsf{P}[Q^c_\ell] \leq \sum_{(j,k,l)\in\mathcal{S}(R_\ell)} \mathsf{P}[Q^c(j,k,l)] && (12)\end{aligned}$$

According to Theorem 4, then for large n, these sums will be dominated by the expression $\Psi(-\sqrt{n\beta})$ where

$$\beta = \min_{\ell=1}^{|V\setminus R|-1} \min_{(j,k,l)\in\mathcal{S}(R_\ell)} \frac{D^2(j,k,l)}{\sigma^2(j,k,l)} \tag{13}$$

For large n, the approximation for $\log P^f$ is asymptotically linear in n with negative slope $\beta/2$. A simple approximation is thus $P^f \approx e^{-n\beta/2}$.

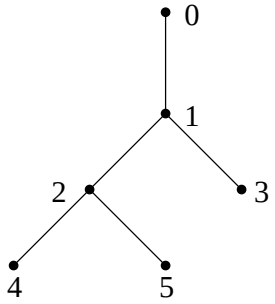


Fig. 2. THE THREE-RECEIVER BINARY TREE.

If we consider the asymptotic regime of small loss and delay, $\|\overline{\alpha}\| \to 0$, from relations (i) and (ii) in Theorem 4 it follows that

$$\min_{(j,k,l)\in\mathcal{S}(R_t)} \frac{D^2(j,k,l)}{\sigma^2(j,k,l)} = \overline{\alpha}(i) + O(\|\overline{\alpha}\|^2), \qquad (14)$$

the minimum being attained, for small enough $\|\overline{\alpha}\|$, where $a(j,k) = i$ and $a(j,l) = f(i)$. Picking the dominant contribution to (11) then $\beta \approx \inf_{i\in V\setminus R} \overline{\alpha}(i)$ yielding $P^f \approx e^{-n\overline{\alpha}(i)/2}$. Thus, in this regime, the probability of correctly identifying the topology is controlled by the smallest loss rate or link utilization.

The above argument can be formalized using Large Deviation theory. However, calculation, of the decay rate appears computationally infeasible, although the leading exponent $\inf_{i\in V\setminus R} \overline{\alpha}(i)$ can be recovered in the small $\|\overline{\alpha}\|$ regime.

B. Comparative Performance of Loss and Utilization-Based Classifiers

As an example we consider the three receiver tree with uniform link probabilities $\alpha_u(k) = \alpha_u$ and $\alpha_l(k) = \alpha_l$; see Figure 2. The topology is correctly inferred when nodes 4 and 5 are grouped together; this requires $A^{(n)}(4,5) < A^{(n)}(4,3)$ and $A^{(n)}(5,4) < A^{(n)}(5,3)$. The argument controlling the misclassification probability is $\beta = D^2(4,5,3)/\sigma^2(4,5,3) = D(5,4,3)^2/\sigma^2(5,4,3)$. We plot this as a function of the common probability α in Figure 3. The curve is approximately linear in $\overline{\alpha}$ for small $\overline{\alpha} = 1-\alpha$, in agreement with (14). As $\overline{\alpha}$ increases, β reaches a maximum at about $\overline{\alpha} = 0.2$ ($\alpha = 0.8$), then decreases to 0. Thus in this homogeneous tree, the misclassification probability is minimized when $\overline{\alpha} \approx 0.2$.

We compare the relative performance of the loss and utilization classifiers in Figure 4, indicating the regions where each of the relevant slopes β_u, β_l is higher. The loss classifier is best when loss rates are higher than about 0.2 (*i.e.,* $\alpha_l \leq 0.8$) or when utilization is high (*i.e.,* low α_u). However, it is outperformed by the utilization classifier when there is low utilization (i.e. high α_u).

C. Performance of JBT

In this case, the analysis of the misclassification probability is complicated by the fact that JBT uses the misclassification estimates to take grouping decisions. Here, to illustrate its modes of misclassification and assess its relative benefit with respect to BT we analyze the performance of JBT in the three

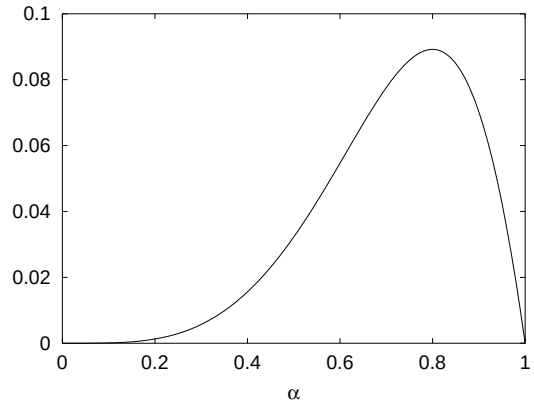


Fig. 3. THREE-RECEIVER TREE. Asymptotic slope of misclassification probability for a single classifier, as function of uniform link probability α

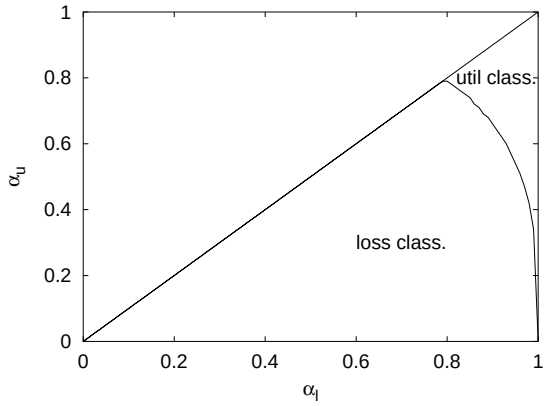


Fig. 4. THREE-RECEIVER TREE. Partition of parameter space(α_l, α_u) where loss or utilization estimator has better performance, i.e. largest asymptotic slope for misclassification probability. Note $\alpha_l < \alpha_u$.

receiver binary tree scenario in Figure 2 with uniform link probabilities. In JBT, the topology is correctly inferred when for the chosen performance measure $A^{(n)}(4,5) < A^{(n)}(4,3)$ and $A^{(n)}(4,5) < A^{(n)}(5,3)$. To keep the complexity manageable, we focus on the first event and assume misclassification occurs when $A^{(n)}(4,5) \geq A^{(n)}(4,3)$, *i.e.,* when $D^{(n)}(4,5,3) < 0$.

The behavior of the classifier is then completely characterized by the bivariate random variable $\boldsymbol{x}^{(n)} = (x_l^{(n)}, x_u^{(n)})$ where $x^{(n)} = \frac{D^{(n)}(4,5,3)}{\sigma^{(n)}(4,5,3)}$. From (6), the misclassification estimate for both performance measures is $P_{453}^{f,(n)} = \Psi(-\sqrt{n}|x^{(n)}|)$; the joint algorithm groups the nodes based on loss information when $|x_l^{(n)}| \geq |x_u^{(n)}|$ and on utilization otherwise (we assume ties are resolved in favor of loss). Misclassification occurs when the chosen performance measure results in grouping the wrong pair; this happens when $|x_l^{(n)}| \geq |x_u^{(n)}|$ and $x_l^{(n)} < 0$ or when $|x_u^{(n)}| > |x_l^{(n)}|$ and $x_u^{(n)} < 0$ which simply amounts to the condition $x_l^{(n)} + x_u^{(n)} \leq 0$. The misclassification probability is then

$$P_j^f := \mathsf{P}[x_l^{(n)} + x_u^{(n)} \leq 0] \qquad (15)$$

Normal Approximation. We now consider the asymptotic behavior of P_j^f. An application of the Delta method (see Chapter 7 of

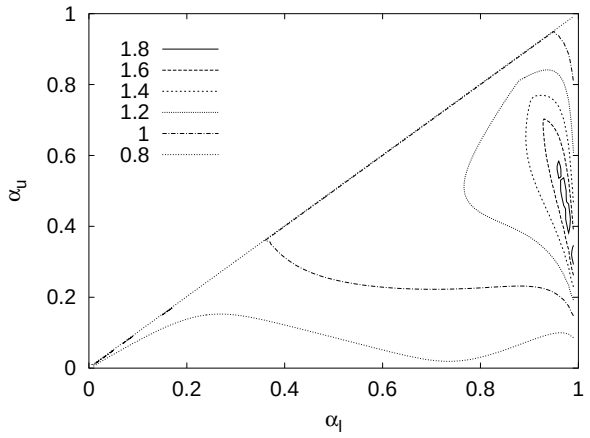


Fig. 5. JOINT CLASSIFIER. Contour plot of the ratio of the (log-scale) misclassification probability asymptotic slope between the joint and best basic classifier.

[17]) shows that as $n \to \infty$, $\sqrt{n}(\boldsymbol{x}^{(n)} - \boldsymbol{x})$, where $\boldsymbol{x} = (x_l, x_u)$, $x = \frac{D(4,5,3)}{\sigma(4,5,3)} = \mathcal{H}(A)$ with continuous $\mathcal{H}$, converges in distribution to a bivariate Gaussian random variable with mean zero and covariance matrix $\sigma_{\boldsymbol{x}} = (\nabla\mathcal{H}(A_l), \nabla\mathcal{H}(A_u)) \cdot \sigma_{A_l,A_u} \cdot (\nabla\mathcal{H}(A_l), \nabla\mathcal{H}(A_u))^T$, where σ_{A_l,A_u} the asymptotic covariance matrix of $\sqrt{n} \cdot (A_l, A_u)$ and $.^T$ denotes the transpose. (σ_{A_l,A_u} can be computed generalizing the approach used in [8] to compute σ_A.)

Therefore, we have the following approximation

$$P_j^f \approx \int_{x_l^{(n)}+x_u^{(n)} \le 0} e^{-\frac{n}{2}(\boldsymbol{x}^{(n)}-\boldsymbol{x})\cdot\sigma_{\boldsymbol{x}}^{-1}\cdot(\boldsymbol{x}^{(n)}-\boldsymbol{x})^T} d\boldsymbol{x} \quad (16)$$

$$\approx e^{-\frac{n}{2}\inf_{x_l^{(n)}+x_u^{(n)}=0}(\boldsymbol{x}^{(n)}-\boldsymbol{x})\cdot\sigma_{\boldsymbol{x}}^{-1}\cdot(\boldsymbol{x}^{(n)}-\boldsymbol{x})^T} \quad (17)$$

where for large n, we consider the leading exponential order. The infimum in (17) is $x_j^2 = (\boldsymbol{x'} - \boldsymbol{x}) \cdot \sigma_{\boldsymbol{x}}^{-1} \cdot (\boldsymbol{x'} - \boldsymbol{x})^T$, where $\boldsymbol{x'} = (x'_l, x'_u) = (x'_l, -x'_l)$ is the tangent point between the line $x_l^{(n)} + x_u^{(n)} = 0$ and the ellipse of the family $(\boldsymbol{x}^{(n)} - \boldsymbol{x}) \cdot \sigma_{\boldsymbol{x}}^{-1} \cdot (\boldsymbol{x}^{(n)} - \boldsymbol{x})^T = a^2$ parameterized in a. Thus, as n goes to infinity we expect the curve $\log P_j^f$ vs. n being asymptotically linear with negative slope $x_j^2/2$. A simple approximation is then $P_j^f \approx e^{-nx_j^2/2}$. Moreover, the minimizing pair $(x_l^{(n)}, x_u^{(n)}) = (x'_l, -x'_l)$ indicates that misclassification most likely occurs by having the two estimated misclassification probabilities equal, loss and utilization yielding two different pairs for grouping, and picking the wrong pair.

To illustrate the results, we study the relative performance of JBT by comparing the asymptotic slope of the logarithm of the misclassification probability x_j^2 with that of the best single classifier. This is computed by considering the leading exponential order approximation $P^f \approx \Psi\left(-\sqrt{n}\frac{D(4,5,3)}{\sigma(4,5,3)}\right) \approx e^{-nx^2/2}$ of the misclassification probability in BT. Figure 5 shows the contour plot of the ratio $\frac{x_j^2}{\max\{x_l^2, x_u^2\}}$ of the (log-scale) asymptotic slopes as function of link characteristics. (α_l, α_u). JBT performs better than either version of BT for a significant range of values (the region within the contour line corresponding to 1). The performance improvement is more pronounced in the region where the loss and utilization classifiers have similar performance (which corresponds to the line separating the two regions in Figure 4) and loss and utilization estimates have low correlation (which occurs when $\alpha_l \gg \alpha_u$). This is not surprising since we expect that: (i) little improvement can be achieved when one classifier significantly outperforms the other; and (ii) strong correlation offsets the benefits of using both loss and utilization estimates.

To show the effect of correlation, consider the case $x_l = x_u$, *i.e.*, when the loss and utilization classifiers have the same performance. In this case, it is easy to verify that $x_j^2 = \frac{2}{1+\rho}x_l^2$, where ρ denotes the coefficient of correlation of $x_l^{(n)}$ and $x_u^{(n)}$. At one extreme, $\rho = 1$ and $x_j^2 = x_l^2$, *i.e.*, $P_j^f = P^f$: we have maximal correlation between the loss and utilization classifiers and JBT cannot provide any performance improvement; at the other extreme, $\rho = 0$ and $x_j^2 = 2x_l^2$, *i.e.*, $P_j^f = P_l^f P_u^f$: we have statistical independence and the probability of misclassification is the product of the two misclassification probabilities.

From Figure 5 we also observe that JBT does not always provide better performance. In this example, we have that under very high or very low utilization the loss and utilization classifiers, respectively, have better performance than JBT. In these cases, because of the high variance of the misclassification probabilities estimates, JBT is likely to mistakenly give preference to the worst performance measure.

VI. EXPERIMENTAL EVALUATION

In this section we evaluate the performance of JBT and compare it with that of BT through two types of simulation. In *model simulations* delay and loss are chosen to follow our statistical model, allowing us to test algorithm performance in the setting on which our analysis is based. *Network simulations*, using the `ns` [13] simulator, test the algorithms in a more realistic setting, where delay and loss are due to queueing delay and buffer overflows at nodes as multicast probes compete with background TCP/UDP traffic.

Model Simulation. We conducted 10000 experiments over randomly generated 15 node binary trees. In Figure 6, we plot the fraction of incorrectly classified topologies as a function of the number of probes for the different classifiers. We considered two regimes: a light load regime with low loss (randomly chosen between 1% and 5%) and utilization (randomly chosen between 10% and 40%), and a heavy load regime with higher loss (randomly chosen between 1% and 20%) and utilization (randomly chosen in between 30% and 80%).

In both cases, the joint classifier dramatically outperform the loss and utilization classifiers with a difference in accuracy already of more than one order of magnitude in accuracy for just 400 probes.

The accuracy of our approach to joint classification lies in that of the misclassification probability estimates. In Figure 6 we also superimposed the mean over the experiments of the topology misclassification probability estimates. From the Figure, we observe that the curves well track the actual slopes, bound from above the actual values and preserve their relative order.

We can use the topology misclassification probability estimate to determine the number of probes required to achieve a

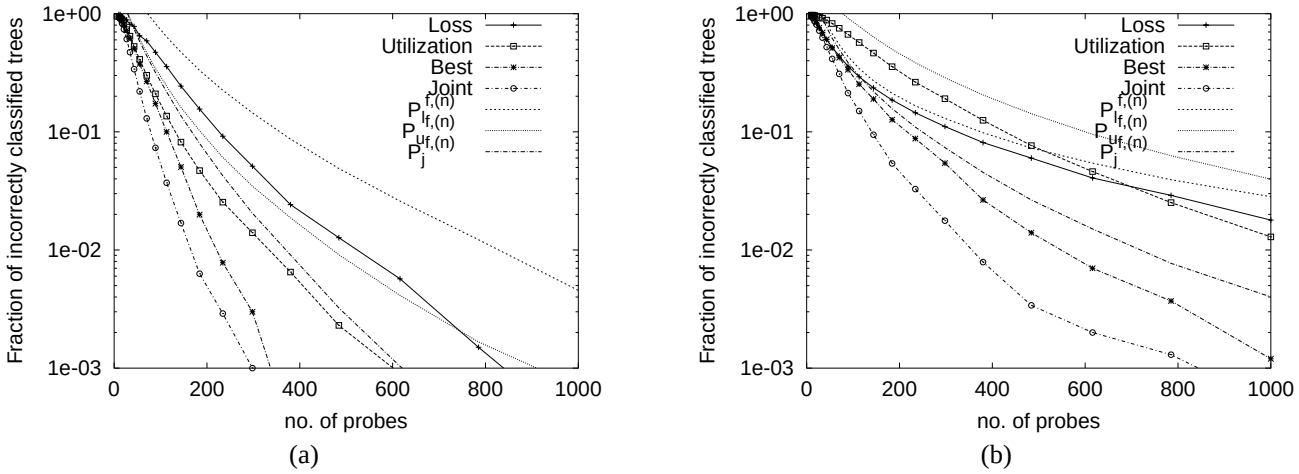


Fig. 6. MODEL SIMULATION. Fraction of incorrectly classified topologies and misclassification estimates for different classifiers as function of number of probes: (a) light load scenario; (b) heavy load scenario.

JBT			
δ	0.05	0.1	0.2
fract. of mis. topologies	0.003	0.008	0.032
average # of probes	145	117	86

BT (loss)			
δ	0.05	0.1	0.2
fract. of mis. topologies	0	0	0.011
average # of probes	415	318	240

TABLE I

ACCURACY OF THE INFERRED TOPOLOGY. FRACTION OF MISCLASSIFIED TOPOLOGIES AND AVERAGE NUMBER OF DISPATCHED PROBES FOR DIFFERENT VALUES OF δ.

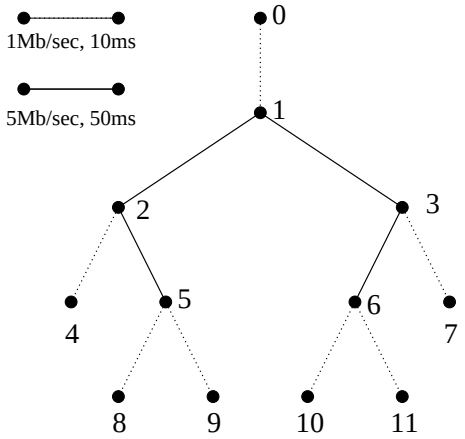


Fig. 7. ns SIMULATION TOPOLOGY.

desired level of accuracy of the inferred topology. The idea is to proceed by dispatching probes until the estimated misclassification probability is below a given threshold δ corresponding to a desired level of accuracy. Thus, for example, to insure a probability of misclassification no greater than 0.05, we send probes until $P^{f,(n)} \leq 0.05$.

We performed 1000 experiments over random generated 15 node binary trees. In each experiment probes were dispatched until the misclassification probability fell below a given threshold δ and we verified whether the inferred topology was correct. For JBT and BT under the light load regime, we summarise the results in Table I where, for different values of δ, we display the average number of probes that were dispatched and the fraction of topologies that were misclassified. Since the estimate bounds from above the misclassification probability, it is no surprise that the fraction of misclassified topologies is well below the chosen threshold. Observe that the number of probes required by JBT is about one third of those required by BT with loss.

Finally, to illustrate the benefit of combining loss and utilization measurements we compare JBT with a simpler approach which simply consists in choosing among the inferred topologies separately computed with the loss and utilization classifiers that with the smallest misclassification probability estimate. Denote $\mathcal{T}_X^{(n)}$ the topology inferred by classifier X, $X \in \{l, u\}$ and $P_X^{f,(n)}$ its estimated probability of misclassification. We select $\mathcal{T}_{best}^{(n)} = \mathcal{T}_Y^{(n)}$, where $Y = \text{argmin}_{X \in \{l,u\}} P_X^{f,(n)}$. In Figure 6 we also superimposed the fraction of times $\mathcal{T}_{best}^{(n)}$ was incorrect. This approach yields more accurate results than either loss and utilization classifiers, yet not as accurate as JBT: the distance from the JBT curve quantifies the significant gain achievable by the adaptive scheme which use both performance measures; the fact the two curves are parallel suggests that misclassification is ultimately dominated by the same event in both cases.

TCP/UDP Network Simulation. The ns simulations used the topology shown in Figure 7. We arranged for some heterogeneity with the interior links having higher capacity (5Mb/sec) and propagation delay (50ms) then at the edge (1Mb/sec and 10ms). Each link is modeled as a FIFO queue with a 20-packets buffer capacity.

The root node 0 generates probes as a 20Kbit/s stream comprising 40 byte UDP packets according to a Poisson process with a mean interarrival time of 16ms. The background traffic comprises a mix of infinite data source TCP connections (FTP) and

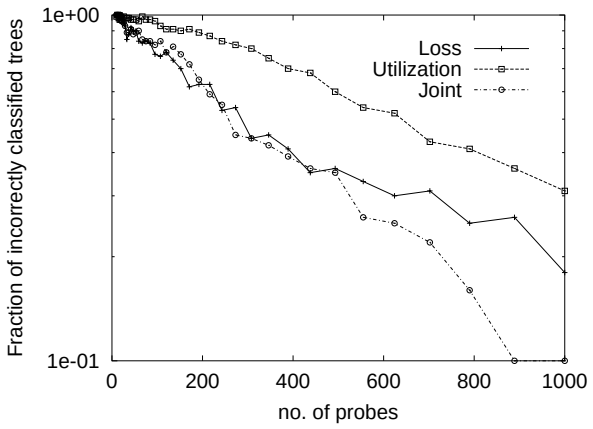


Fig. 8. ns SIMULATION. Fraction of incorrectly classified topologies for different classifiers as function of the number of probes.

exponential on-off sources using UDP. Averaged over the different simulations, the link loss ranges between 1% and 13% and link utilization ranges between 10% and 88%.

Figure 8 plots the fraction of incorrectly identified topologies over 100 simulations. The relative accuracy among the different classifiers is in good agreement with the results from the model simulations. Performance of the utilization and joint classifiers are somewhat inferior due to: (i) wide spread of link utilization values among the different links; (ii) presence of spatial correlation among probe delays. In the simulations, probes are more likely to experience similar level of congestion on consecutive or sibling links than dictated by the modes independence assumption. We calculated the off-diagonal elements of the correlation matrix of the actual link delays. The mean was 0.021 and the maximum 0.17. Despite correlation affected its accuracy, JBT shows, albeit reduced, performance gain over BT.

In the simulations we also observed the presence of short-term temporal correlation among successive probes that encountered the same congestion events. This does not affect estimator consistency, although the convergence rate may be slowed.

VII. CONCLUSIONS

In this paper we have presented an algorithm for the inference of the multicast tree topology from end-to-end measurements. The algorithm combines different performance measures and reconstruct the tree by adaptively choosing that which insures the best accuracy. This is accomplished by a careful enumeration of all the possible erroneous decisions and by estimation of their probability. These estimates in turn can be used to determine the number of probe packets to achieve a desired level of accuracy.

We investigated the statistical properties of the algorithm and showed that it is consistent. Analysis of a simple scenario showed that it can significantly outperform any of the algorithms previously considered. We also used simulation to evaluate its accuracy and found out that, in general, it required many fewer probes to correctly identify the topology than other approaches. ns experiments showed that spatial correlation negatively affects its accuracy. We believe that diversity of traffic in real networks makes large and long lasting correlation unlikely. We are currently investigating the effect of correlation on the accuracy of topology inference algorithms; this is part of a more general effort to characterize network traffic correlation and its effects on end-to-end measurements based inference.

Acknowledgment. We thank Don Towsley for useful comments and suggestions.

APPENDIX

The proof of Theorem 1 is based on the following result. We will find it useful to identify a subset S of V as a **stratum** if $\{R(k) : k \in S\}$ is a partition of R.

Lemma 1: Let S be a stratum. Then,

(i) a pair of nodes $\{j,k\} \subseteq S$ are siblings if and only if

$$A(j,k) < \min_{\{j',k'\} \subset S : |\{j',k'\} \cap \{j,k\}| = 1} A(j',k'); \qquad (18)$$

(ii) if $\{j,k\} \subseteq S$ are such that

$$A(j,k) = \min_{\{j',k'\} \subset S} A(j',k') \qquad (19)$$

then $\{j,k\}$ are sibling;

(iii) if $\{j,k\} \subseteq S$ is a pair of sibling nodes, then $(S \setminus \{j,k\}) \cap \{a(j,k)\}$ is a stratum.

Proof:. Observe first that by definition of stratum, if $j \in S$, then no ancestor or descendent of j can belong to S. (i) the *only if* part follows from the observation that if j and k are sibling, then $a(j,k) \prec a(j,\ell), a(\ell,k)$ for any $\ell \in S \setminus \{j,k\}$ which implies $A(j,k) < A(j,\ell), A(\ell,k)$. For the *if* part assume that $\{j,k\} \subset S$ satisfies (18) and suppose j and k are not siblings. Let ℓ be the sibling of j. Then, $\ell \notin S$ since, if $\ell \in S$, $a(j,\ell) \prec a(j,k)$ implies $A(j,\ell) < A(j,k)$, contradicting (18). Thus, since S is a stratum, there is a set of nodes $T = \{t_1, \ldots, t_{n_\ell}\} \subseteq V(\ell) \cap S$ such that $\cup_{i=1}^{n_\ell} R(t_i) = R(\ell)$ since otherwise $\cup_{i \in S} R(i)$ would not cover R. Now either $k \in T$ or $k \notin T$. But $k \in T$ implies that $a(i,k) \prec a(j,k)$, $i \in T$ so that $A(i,k) < A(j,k)$ contradicting (18) while $k \notin T$ implies that $a(j,i) \prec a(j,k)$, $i \in T$ so that again $A(j,i) < A(j,k)$ contradicts (18). Therefore j and k are siblings. (ii) then is an immediate consequence of (i) and (iii) follows immediately from the definition of stratum. ∎

Proof of Theorem 1. It suffices to observe that in DBT, at the beginning of each iteration, R' is a stratum; therefore, the pair of nodes which minimizes $A(.,.)$ is always a pair of sibling nodes. This property holds before the first loop (R is a stratum), and (ii) and (iii) of Lemma 1 ensure it holds subsequently. ∎

Proof of Theorem 3. Since $A^{(n)}(j,k)$ converges almost surely to $A(j,k)$, then, with probability 1, for all sufficiently large n, the relative ordering of the $A^{(n)}(j,k)$ is the same as that of $A(j,k)$ (which observe can be different for loss and utilization). Then, it suffices to observe that for all sufficiently large n, the two pairs of nodes which minimize $A_l(.,.)$ and $A_u(.,.)$ are both siblings pairs provided R' is a stratum. This property holds before the first loop (R is a stratum), and (iii) of Lemma 1 insure it holds subsequently, irrespectively of the actual pair of nodes selected for grouping. Then the last two statements directly follows from standard results.

REFERENCES

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", *IEEE Communications Magazine*, May 2000.

[2] CAIDA: Cooperative Association for Internet Data Analysis. For more information see `http://www.caida.org`

[3] R. Caceres, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics", *IEEE Trans. on Information Theory*, November 1999.

[4] R. Caceres, N.G. Duffield, J.Horowitz F. Lo Presti and D. Towsley, "Statistical Inference of Multicast Network Topology", *Proc. IEEE Conference on Decision and Control*, Phoenix, AZ, Dec 1999.

[5] Cooperative Association for Internet Data Analysis, "Internet Measurement Efforts," `http://www.caida.org/Tools/taxonomy.html#InternetMeasurement`

[6] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", *Proc. IEEE Infocom 2000*, Tel Aviv, March 2000.

[7] N.G. Duffield, J.Horowitz, F. Lo Presti, D. Towsley, "Multicast Topology Inference from End-to-End Measurements", *to appear in Proc. of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Monterey, CA, Sept 2000.

[8] N.G. Duffield, J.Horowitz, F. Lo Presti and D. Towsley, "Multicast Topology Inference from Measured End-to-End Loss", submitted for publication.

[9] Felix: Independent Monitoring for Network Survivability. For more information see `ftp://ftp.bellcore.com/pub/mwg/felix/index.html`

[10] IPMA: Internet Performance Measurement and Analysis. For more information see `http://www.merit.edu/ipma`

[11] F. Lo Presti, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[12] D. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", *RFC 1305*, Network Information Center, SRI International, Menlo Park, CA, Mar. 1992.

[13] `ns` - Network Simulator. For more information see `http://www-mash.cs.berkeley.edu/ns/ns.html`

[14] S. Paul at al. "Reliable Multicast Transport Protocol (RMTP)", *IEEE JSAC* Vol. 15, No. 3, pp. 407-421, April 1997.

[15] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, Vol. 36, No. 8, pp. 48-54, August 1998.

[16] S. Ratnasamy & S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", *Proc. IEEE Infocom'99*, New York, NY (1999)

[17] M. Schervish, "Theory of Statistics", Springer, New York, 1995.

[18] Surveyor. For more information see `http://io.advanced.org/surveyor/`

Inferring Link Loss Using Striped Unicast Probes

N.G. Duffield[†] F. Lo Presti[†,§] V. Paxson[‡] D. Towsley[§]

[†]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
{duffield, lopresti}@research.att.com

[‡]ACIRI
International Computer Science Institute
Berkeley, CA 94704, USA
vern@aciri.org

[§]Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
towsley@cs.umass.edu

Abstract— **In this paper we explore the use of end-to-end unicast traffic as measurement probes to infer link-level loss rates. We leverage off of earlier work that produced efficient estimates for link-level loss rates based on end-to-end multicast traffic measurements. We design experiments based on the notion of transmitting stripes of packets (with no delay between transmission of successive packets within a stripe) to two or more receivers. The purpose of these stripes is to ensure that the correlation in receiver observations matches as closely as possible what would have been observed if the stripe had been replaced by a notional multicast probe that followed the same paths to the receivers. Measurements provide good evidence that a packet pair to distinct receivers introduces considerable correlation which can be further increased by simply considering longer stripes. We then use simulation to explore how well these stripes translate into accurate link-level loss estimates. We observe good accuracy with packet pairs, with a typical error of about 1%, which significantly decreases as stripe length is increased to 4 packets.**

I. Introduction

A. Motivation

As the Internet grows in size and diversity, its internal performance becomes ever more difficult to measure. Any one organization has administrative access to only a small fraction of the network's internal nodes, whereas commercial factors often prevent organizations from sharing internal performance data.

One promising technique that avoids these problems, *Multicast Inference of Network Characteristics* (MINC), uses end-to-end multicast measurements to infer link-level loss rates and delay statistics by exploiting the inherent (and well characterized) correlation in performance observed by multicast receivers. These measurements do not rely on administrative access to internal nodes since the inference can be calculated using only information recorded at the end hosts.

The key intuition for inferring packet loss is that the arrival of a packet at a given internal node can be directly inferred from the packet's arrival at one or more receivers reached from the source by paths through that node; if it makes it to the receivers, it must have made it to the node. Conditioning on arrival at a descendent, we can determine the probability of successful transmission to and beyond the given node. Efficient inferencing algorithms are given in [2] for loss, [15] for delay distributions, [7] for delay variances, and [3] for inferring the logical multicast tree topology itself.

Although significant advances have been made in the use of multicast measurements for inferring internal network behavior, it suffers from two serious deficiencies. First, there remain significant portions of the Internet that do not support network-level multicast. Second, the internal performance observed by multicast packets often differs significantly from that observed by unicast packets. This is especially serious given that unicast traffic constitutes far and away the largest portion of the traffic on the Internet. Thus there is a need for techniques based on end-to-end unicast measurements. This poses a significant challenge because unicast measurements do not exhibit the well-behaved correlation exhibited by multicast. Thus, the challenge addressed in this paper is that of developing unicast-based measurement techniques that create sufficient correlation to yield fruitful inference.

B. Contribution

In this paper we adapt the multicast inference techniques proposed in [2] to perform inference of internal network characteristics from unicast end-to-end measurements. The data for the inference comprises measured end-to-end loss of unicast probes sent from a source to a number of destinations. This is used to infer the loss and delay characteristics of each logical link of the source tree joining the source to the destinations, i.e., of the composite paths between its branch points.

The idea is to construct composite probes of unicast packets whose collective statistical properties closely resemble those of a multicast packet. We shall speak of **striping** a group of unicast packets across a set of destinations. This entails dispatching the packets back-to-back

This work was supported in part by DARPA and the AFL under agreement F30602-98-2-0238

from a source, each packet potentially having a different destination address. Our premise is that when the duration of network congestion events exceeds the temporal width of the stripe, packets should have very similar experience of the network upon traversing common portions of the paths to their destinations. If the experiences were identical, the packets from a stripe that attempt to traverse a given link would either all be lost, or encounter identical delay. Hence the packet loss and delays on a given link would be perfectly correlated within a stripe; the composite probe would have the same statistical properties as a notional multicast packet that followed the same source tree. In this case the methods of [2], [7], [15] could be applied immediately to infer the per link loss and delay statistics of the logical source tree.

However, correlations within stripes may be less than perfect in practice. This is because congestion events may not affect packets uniformly, subjecting stripes to dispersion as they travel through a network. Some mechanisms by which this can happen are the following. Packet loss will not be uniform during loss events that are narrower than the stripe, or those that start or stop while the stripe is in progress. Furthermore, delays will vary due to interleaving of background traffic, e.g., when moving from a low to a high capacity link. Although such effects should be small for sufficiently narrow stripes, they will be cumulative. Packet-dropping on the basis of Random Early Detection (RED) [9] is another mechanism by which packet loss may become decorrelated. It remains to be seen whether this mechanism will be widely deployed in communications networks. On the other hand, the use of RED to merely mark packets will not break correlations.

This motivates four strands of work in this paper:
(i) determining the magnitude of imperfect correlations through experiments on real networks;
(ii) calculating their likely impact on the accuracy of inference methods that assume perfect correlations;
(iii) adopting measurement procedures that reduce the impact of imperfect correlations;
(iv) verifying the accuracy of the approach in simulations.

We extend the packet loss model of [2] by incorporating an additional parameter for each link that describes the correlation of loss between different packets of the same stripe. This is done for binary stripes, i.e., those comprising two packets with different destination addresses. These additional parameters cannot themselves be determined by end-to-end measurements, at least not without additional assumptions relating them to each other, or to the existing loss rate parameters. These calculations show that the error in using the loss estimator from [2] is small provided that the conditional probability of *loss* of one packet in the stripe given *transmission* (i.e., non-loss) of the other, is small compared with the marginal loss rate in the stripe. This is a condition that we will verify, at least for end-to-end paths, through measurement.

By constructing appropriate stripes of composite probes and selecting subsets of these probes for inference, we are able to enhance correlations within data used for inference. This is possible when packet transmissions are correlated in the sense that a given packet in a stripe is more likely to be transmitted across a given link when other packets within the stripe are known to have been transmitted across the link. By conditioning on the measurable event that nearby packets have been transmitted end-to-end, we can raise the likelihood of transmission of a given packet to an intermediate node closer to one. By sending the stripe packets to diverse addresses, we can infer the properties of internal network paths from the measurements.

The rest of the paper is as follows. In Section II we formulate the stripe method, first for the binary tree of depth two, and then for general trees. We specify a family of different striping methods. We specify the required correlation assumption between packet transmissions within stripes, and show that it can be used to construct a hierarchy amongst the various striping methods; in particular we establish an order relation for the degree of correction each method gives to the bias caused by imperfect correlations.

We use two experimental approaches to evaluate the proposed method. In Section III we use end-to-end measurement on the National Internet Measurement Infrastructure (NIMI) [19] to gather data from a diverse set of Internet paths. We transmitted stripes between pairs of end-hosts and verified that their packet loss statistics were consistent with the correlation assumptions that underlie the method. (These stripes were different from those defined above, since all packets in the stripes were sent to the same destination; see Section III-A for discussion of this approach.) We also estimated the likely accuracy that would be obtained by stripe-based inference in the actual network.

We support this work in Section IV using network level simulation with `ns` [17]. By instrumenting the simulation we can trace the behavior of packets in the network interior. This allows us first to study the correlation properties of packets within stripes as they are transmitted across individual links in the network (rather than just the end-to-end properties), and second to compare the inferred link loss rates with actual link loss rates. For the most accurate choice of striping method we find the typical absolute error in loss rate inference to be below 1%. We conclude in Section V.

C. Related Work

There exist several tools and methodologies for characterizing link-level behavior from end-to-end unicast measurements. One of the first methodologies focuses on identifying the bottleneck bandwidth on a unicast route. The key idea is that, in an uncongested network, two packets (packet pair) sent back-to-back will arrive at the receiver with a spacing that is inversely proportional to the lowest link bandwidth on the path. This was noted by Jacobson as leading to TCP's "self-clocking" behavior [10], and formally analyzed by Keshav [12]. Carter and Crovella then developed a tool to apply the technique [4], which has since been refined in [13], [18]. Although these methodologies focus on a metric other than loss rate, they are based on the same idea, namely to send packet pairs (or stripes) so as to introduce correlation in a controlled manner.

In [5], the authors use end-to-end measurements of packet pairs in a tree connecting a single sender to several receivers. Experiments consist of a number of packet pairs where the packets are sent to different receivers so that all pairs of receivers are covered. The metrics of interest are success probabilities of all links in the tree. As the second packet in a pair may not see the same loss behavior as the first over the common path, conditional success probabilities are introduced as unknown nuisance variables. Given an *a priori* distribution for these two sets of parameters, the authors then use a Bayesian network approach to determine *a posteriori* distributions and, from these, estimates of the link transmission probabilities. Preliminary results on the method reported in the paper show promise. Our approach differs from the approach in [5] in that we consider a more general form of striping scheme which results in significantly higher correlation. Thus we are able to continue to rely on the maximum likelihood estimates derived for the multicast case.

Last, `pathchar` [6], [11] triggers ICMP messages at successive routers on a unicast path in order derive link bandwidth, round trip link loss rate, and round trip link delay statistics. It accurately estimates link bandwidth provided that it is low. It has not been well validated in the case of losses and delays. Moreover, it requires considerable time to converge and loses accuracy with asymmetric round trip paths.

II. Inference Methodology

A. Models for Trees, Stripes, and Packet Loss

We first develop the framework in which to describe the propagation of stripes of unicast packets through the network. We represent the underlying physical network as a graph $G_{\rm phys} = (V_{\rm phys}, L_{\rm phys})$ comprising the physical nodes $V_{\rm phys}$ (e.g. routers and switches) and the links $L_{\rm phys}$ between them. We consider a single source of probes $0 \in V_{\rm phys}$ and a set of receivers $R \subset V_{\rm phys}$. We assume that the set of paths from 0 to each $r \in R$ is stationary and form a tree $\mathcal{T}_{\rm phys}$ in $(V_{\rm phys}, L_{\rm phys})$; thus two such paths never intersect again once they have diverged. We form the logical source tree $\mathcal{T} = (V, L)$ whose vertices V comprise 0, R and the branch points of $\mathcal{T}_{\rm phys}$. The link set L contains the link (j, k) if one or more of the probe paths in $\mathcal{T}_{\rm phys}$ pass through j then k without encountering another element of V in between. Where applicable, denote by $f(k) \in V$ the parent of $k \in V$. We write $j \succ k$ if j is an ancestor of k in $\mathcal{T}$.

We will use the notation $\langle r_1, \ldots, r_{d_0} \rangle$ to refer to a stripe comprising packets dispatched to destination nodes in order $r_1, \ldots, r_{d_0}$. We describe the progress of the stripe in $\mathcal{T}$ by the variables $X_k(d)$, taking the value 1 if packet d reaches node k, and zero otherwise. Note $X_{r_d}(d) = 1$ iff packet d reaches its destination node. (We do not label packets by their destination since we consider stripes with repeated destinations).

We will find it useful to have a notation describing composite events at sets of receivers. For $D \subset D_0 = \{1, \ldots, d_0\}$ define the binary variable

$$Z_D = \prod_{d \in D} X_{r_d}(d). \tag{1}$$

Thus $Z_D = 1$ if all packets in D reach their destinations, and 0 otherwise. We will find it convenient to write $Z_{\{d_1, \ldots, d_m\}}$ as $Z_{d_1 \ldots d_m}$.

We specify a loss model for the stripes. We assume that losses are independent between different stripes, and for packets of the same stripe on different links. For each $k \in V$ let $D(k) \subset D_0$ be the set of packets that successfully reach (and therefore transit across) k. For $D \subset D(k)$ let $\alpha_k(D)$ denote the probability that all packets in D are transmitted to node k, conditioned upon having reached the parent node $f(k)$. We do not assume that the marginal probabilities $\alpha_k(d)$ are equal for all $d \in D(k)$. For disjoint subsets $D, D' \subset D(k)$ we write as $\beta_k(D|D')$ the conditional probability that packets in D are successfully transmitted across link k, given that those in D' are successfully transmitted, all packets having reached the parent node $f(k)$. This is expressed in terms of the probabilities α_k as

$$\beta_k(D|D') = \alpha_k(D \cup D')/\alpha_k(D'). \tag{2}$$

With perfect correlations the various β_k would be 1. The multicast loss model of [2] is statistically equivalent to the special case $\beta_k(D|D') = 1$ and hence $\alpha_k(d)$ all equal some α_k.

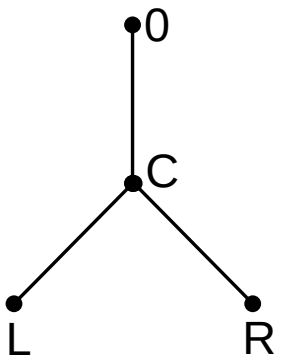


Fig. 1. TWO-LEAF TREE

For a given link and stripe width, we expect the structure of the probabilities α, β to depend on the times between successive packets. For example, if the packets are widely separated, then the marginal probabilities $\alpha_k(d)$ will be equal (or nearly so) while the conditional probabilities β will be close to the marginal probabilities α. Here, we concentrate on the other extreme with back-to-back packets in order to make β close to 1. In this paper we focus on estimating transmission probabilities for the first packet in a stripe. We note however that marginal transmission probabilities can depend on the position of a packet within a stripe, particularly when the stripe width is not negligible compared with buffer sizes. However, our methods can be adapted to focus on other packets within the stripe. This could be useful if it is desired to infer transmission probabilities for packets in traffic bursts.

B. Inference with Binary Stripes on the Two-Leaf Tree

We first investigate the performance of the inference algorithms from [2] under imperfect correlations. We start with the two-leaf tree shown in Figure 1, having leaf nodes L and R with common parent C whose own parent is the root 0. Consider the binary stripe $\langle \mathrm{L}, \mathrm{R} \rangle$. The link probabilities are related to the probabilities of leaf events as follows:

$$\frac{\mathsf{E}Z_1\mathsf{E}Z_2}{\mathsf{E}Z_{12}} = \alpha_{\mathrm{C}}(1)/\beta_{\mathrm{C}}(1|2) \tag{3}$$

$$\frac{\mathsf{E}Z_{12}}{\mathsf{E}Z_2} = \alpha_{\mathrm{L}}(1)\beta_{\mathrm{C}}(1|2), \qquad \frac{\mathsf{E}Z_{12}}{\mathsf{E}Z_1} = \alpha_{\mathrm{R}}(2)\beta_{\mathrm{C}}(2|1),$$

where Z_D is as defined in (1). This is because, e.g., $\mathsf{E}Z_{12} = \alpha_{\mathrm{C}}(12)\alpha_{\mathrm{L}}(1)\alpha_{\mathrm{R}}(2) = \alpha_{\mathrm{C}}(1)\beta_{\mathrm{C}}(2|1)\alpha_{\mathrm{L}}(1)\alpha_{\mathrm{R}}(2)$, with similar expressions for $\mathsf{E}Z_1$ and $\mathsf{E}Z_2$. With perfect correlations, $\beta_{\mathrm{C}} = 1$, and hence the α are uniform across the stripe and may be recovered directly from the leaf probabilities. These expressions can then be used to estimate the α from the leaf events $Z^{(i)}$ associated with multiple identical stripes $i = 1, 2, \ldots n$. To form the estimates we first replace each expectation in (3) by the corresponding empirical mean, defined here in general:

$$\widetilde{Z}_D = n^{-1} \sum_{i=1}^{n} Z_D^{(i)}. \tag{4}$$

Taking $\beta_{\mathrm{C}} = 1$ then yields the estimates

$$\widehat{\alpha}_{\mathrm{C}} = \widetilde{Z}_1\widetilde{Z}_2/\widetilde{Z}_{12},\ \widehat{\alpha}_{\mathrm{L}} = \widetilde{Z}_{12}/\widetilde{Z}_2,\ \widehat{\alpha}_{\mathrm{R}} = \widetilde{Z}_{12}/\widetilde{Z}_1. \tag{5}$$

This is effectively the estimator from [2] applied to the two-leaf tree.

With imperfect correlations, β_{C} cannot be recovered independently from the leaf expectations. The model is not identifiable; this was also observed in [5]. Since $\beta_{\mathrm{C}} \leq 1$, estimation via (5) is biased, overestimating α_{C} and underestimating α_{L} and α_{R}.

C. Enhancing Stripe Correlations

The uncertainty over the values of the β undermines confidence in using (5) directly. We now propose a modified striping scheme scheme for which the effective value of the β is closer to 1. To glimpse the idea behind this, observe that for the stripe $\langle \mathrm{L}, \mathrm{R} \rangle$ with perfect correlations, $\mathsf{E}Z_{12}/\mathsf{E}Z_2$ (defined as the conditional probability for the first packet of the stripe to reach L given that its second packet reaches C) is actually equal to the probability of transmission of a packet along the link (C, L), conditional upon reaching C. This is because packet 2 must have been present at C if present at R. With imperfect correlations, packet 1 may not have been also present at C, leading to underestimation of α_{L}. Our remedy for this is to use longer stripes, conditioning on an event at R which makes it more likely that packet 1 was present at C.

The simplest example of such a stripe is the **three-packet stripe** $\langle \mathrm{L}, \mathrm{R}, \mathrm{R} \rangle$. Provided that transmission of packets within the stripe is strongly correlated (in a sense we make precise below) we expect it to be more likely that packet 1 reaches C, upon reception of packets 2 and 3 at receiver L, rather than reception of packet 2 alone. We formalize the required notion of correlation in Definition 1 below.

Upon replacing the reception of packet 2 with the reception of packets 2 and 3, the analogs of the first and second relations in (3) are

$$\frac{\mathsf{E}Z_1\mathsf{E}Z_{23}}{\mathsf{E}Z_{123}} = \frac{\alpha_{\mathrm{C}}(1)}{\beta_{\mathrm{C}}(1|23)},\ \frac{\mathsf{E}Z_{123}}{\mathsf{E}Z_{23}} = \alpha_{\mathrm{L}}(1)\beta_{\mathrm{C}}(1|23). \tag{6}$$

The parameters α_{C} and α_{L} are estimated by $\widetilde{Z}_1\widetilde{Z}_{23}/\widetilde{Z}_{123}$ $\widetilde{Z}_{123}/\widetilde{Z}_{23}$ respectively; α_{R} can be estimated similarly using the complementary stripe $\langle \mathrm{R}, \mathrm{L}, \mathrm{L} \rangle$. Comparing with

(5) we observe that these estimates introduce less bias than those from two-packet stripes provided that $\beta_{\rm C}(1|2,3) > \beta_{\rm C}(2|1)$. This is the case provided that transmissions within a stripe satisfy the following correlation property.

Definition 1: We say that stripe transmission at a node k is **coalescent** if for each stripe $\langle r_1, \ldots, r_d \rangle$ routed through k, and disjoint $D, D' \subset D(k)$,

$$\beta_k(D|D') \geq \beta_k(D|D'') \text{ for all } D'' \subset D'. \qquad (7)$$

Coalescence is a correlation property. It states that a given set of packets D is more likely to be transmitted on a link, the more other packets from the stripe have been transmitted. We will investigate the coalescence properties of real network traffic in Section III.

With coalescence, whenever we add packets to the conditioning event, the effect is to decrease the estimate of $\alpha_{\rm C}$ and to increase the estimate of $\alpha_{\rm L}$ or $\alpha_{\rm R}$. Thus, we can counteract the bias in the two-leaf stripe, evident from (3), by using wider stripes.

Theorem 1: Assume transmission is coalescent on the two-leaf tree and consider a stripe $\langle D({\rm C}) \rangle$ and two disjoint subsets D, D' of $D({\rm C})$ such that packets in D have destination ${\rm L}$ and packets in D' have destination ${\rm R}$. Then for any $D'' \subset D'$,

$$\frac{\mathsf{E} Z_{D \cup D'}}{\mathsf{E} Z_{D'}} \geq \frac{\mathsf{E} Z_{D \cup D''}}{\mathsf{E} Z_{D''}}. \qquad (8)$$

The inequality (8) captures the effect that extending the stripe reduces the estimate of the transmission rate $\alpha_{\rm C}$ and so counteracts the bias due to $\beta_{\rm C} < 1$.

Proof: $\mathsf{E} Z_{D \cup D'} = \beta_{\rm C}(D|D')\alpha_{\rm C}(D')\ \alpha_{\rm L}(D)\alpha_{\rm R}(D')$ while $\mathsf{E} Z_{D'} = \alpha_{\rm C}(D')\alpha_{\rm R}(D')$. Hence $\mathsf{E} Z_{D \cup D'}/\mathsf{E} Z_{D'} = \beta_{\rm C}(D|D')\alpha_{\rm L}(D) \geq \beta_{\rm C}(D|D'')\alpha_{\rm L}(D) = \mathsf{E} Z_{D \cup D''}/\mathsf{E} Z_{D''}$. ■

Example: the 4-packet stripe. Theorem 1 suggests we can further reduce bias by lengthening the stripe length. Consider, for instance, the stripe $\langle {\rm L}, {\rm R}, {\rm R}, {\rm R} \rangle$ and compare its estimation properties with those of its substripes $\langle {\rm L}, {\rm R}, {\rm R} \rangle$ and $\langle {\rm L}, {\rm R} \rangle$. By Theorem 1 we have the following ordering between the functional on which estimates of $\alpha_{\rm C}$ are based in each case:

$$\frac{\mathsf{E} Z_1 \mathsf{E} Z_{234}}{\mathsf{E} Z_{1234}} \leq \frac{\mathsf{E} Z_1 \mathsf{E} Z_{23}}{\mathsf{E} Z_{123}} \leq \frac{\mathsf{E} Z_1 \mathsf{E} Z_2}{\mathsf{E} Z_{12}}. \qquad (9)$$

The estimators are obtained by replacing each $\mathsf{E} Z$ by the corresponding empirical mean $\widetilde{Z}$ from n stripes. By the Law of Large Numbers, the same inequalities hold for the estimates with probability 1 as n grows to infinity.

D. Extension to General Trees

We describe estimators that extend the foregoing method to treat general logical source trees, i.e., trees in which the depth and branching ratio can be greater than 2. Consider first the case of a depth 2 tree with an arbitrary number of leaves. One approach is to stripe across all receivers and then to adapt the estimator from [2] for nodes with arbitrary numbers of offspring in order to estimate the link probabilities. A potential problem with this approach is that the statistical properties of stripes may not reflect those of general traffic if their width is not negligible compared with buffer sizes. For the same reason, variation of stripe width within a single set of measurements may introduce non-uniform bias into the link probability estimates, depending on the local branching ratio. Instead, here we focus on combining inference from fixed-width stripe measurements on embedded subtrees.

Consider an arbitrary tree with leaf set R. For each node k let $R(k)$ denote the subset of leaves descended from k. Let $Q(k)$ denote the set of ordered pairs of nodes in $R(k)$ descended through different children of k. For each $({\rm R}_1, {\rm R}_2) \in Q(k)$, consider the embedded two-leaf binary tree spanned by the nodes $0, k, {\rm R}_1, {\rm R}_2$. By combining estimates from measurements of stripes down each such tree, we shall estimate the characteristics of the common path from 0 to k.

Each stripe will follow the same pattern. We fix a template for a stripe of d_0 packets by partitioning $\{1, \ldots, d_0\}$ into two sets D_1, D_2. For each ordered pair $({\rm R}_{i_1}, {\rm R}_{i_2})$ of distinct receivers in $R(k)$ we form a stripe that sends packets in positions in D_1 to ${\rm R}_{i_1}$ and packets in positions in D_2 to ${\rm R}_{i_2}$. More formally, this is the stripe $S(i_1, i_2) = \langle r_1, \ldots, r_{d_0} \rangle$ where $r_d = {\rm R}_{i_k}$ when $d \in D_k$.

The relation between the leaf probabilities and the transmission probabilities on the composite path from 0 to k are expressed through

$$\frac{\mathsf{E} Z_{D_1} \mathsf{E} Z_{D_2}}{\mathsf{E} Z_{D_1 \cup D_2}} = A_k(D_1)/B_k(D_1|D_2). \qquad (10)$$

where $A_k = \prod_{j \succeq k} \alpha_j$ and $B_k = \prod_{j \succeq k} \beta_j$. For each non-leaf and non-root node k, each pair $(i, j) \in Q(k)$, the measurements with n stripes of type $S(i, j)$ thus gives rise to an estimate

$$\widehat{A}_k^{i,j} = \frac{\widetilde{Z}_{D_1} \widetilde{Z}_{D_2}}{\widetilde{Z}_{D_1 \cup D_2}}. \qquad (11)$$

In the experiments described in this paper we combine all possible estimates through their arithmetic mean

$$\widehat{A}_k = \#Q(k)^{-1} \sum_{(i,j) \in Q(k)} \widehat{A}_k^{i,j}. \qquad (12)$$

For leaf nodes k take $\widehat{A}_k$ as the measured transmission probability over all stripes of packets to k, and set $\widehat{A}_0 = 1$

by convention. The link probability estimates are then expressed as quotients

$$\widehat{\alpha}_k = \widehat{A}_k / \widehat{A}_{f(k)}, \quad k \neq 0. \tag{13}$$

E. Sampling and Statistical Issues

Earlier in this section we proposed using wider stripes as a way of counteracting the inherent bias in using estimators that do not take explicit account of the imperfect correlations between stripe packets. We now make a number of further observations of the statistical implications of using the stripe approach.

First, increasing the stripe width while keeping the total number of packets sent constant increases the variance of the estimates. This is because the number of stripes sent is in inverse proportion to their width.

Second, network characteristics may not be uniform across a stripe e.g., if stripe width is comparable in size to that of a buffer. Here we focused on estimating transmission probabilities for the first packet; other templates could direct attention to other positions. We note that if marginal transmission rates are highly heterogeneous across different positions in a stripe, then the assumption of independent packet loss on different links may not hold. This is because its expected loss rate of a packet at a given node can depend on the occurrence of losses closer to the source of packets in earlier stripe positions. These cause the packet to advance its position in the stripe and consequently experience a different loss rate.

Third, there is a phenomenon during TCP slow start that can lead to every other or every third packet being lost. Once TCP increases its window enough to "fill the pipe," which corresponds to transmitting at the bottleneck rate, then the next set of acknowledgements effectively increases the sending rate by either a factor of two (if the receiver acknowledges every incoming packet) or a factor of 1.5 (if the receiver uses the common "ack every other" policy). If the bottleneck buffer is full at this point, then either every other or every third packet will be lost at the bottleneck due to the mismatch between the bottleneck rate and the higher sending rate. See Figure 2 of [8] for an illustration. Accordingly, there may be buffer-filling patterns present in the network that impart particular loss patterns on the elements of a stripe. The prevalence of the "slow start" pattern will depend on how often TCP connections in slow start dominate the consumption of buffer space at the bottleneck link.

Fourth, we have observed that imperfect correlations at a node bias inference for parent and child links in opposite directions. Hence bias is a second order effect spatially, depending not on the absolute loss correlation, but rather on the manner in which it changes from node to node in the network. In the special case of the probabilities α, β being uniform over all links, imperfect correlations actually leave the estimates (5) *unbiased* for internal links (i.e. all those except the leaf links and root link), though this special case seems highly unlikely in practice.

Fifth, the analysis of estimator variance for multicast inference carries over when $\beta \approx 1$. We refer the reader to [2] for details. Here we mention that in a regime for which all loss rates $\overline{\alpha}_k = 1 - \alpha_k$ are close to zero, the estimator $\widehat{\alpha}_k$ has variance which behaves as $n^{-1}\left(\overline{\alpha}_k + \|\overline{\alpha}\|^2\right)$, asymptotically for large numbers n of probes. To leading order, this form is independent of topology.

III. Network Experiments

The estimation techniques described in Section II rely on conditional probabilities of packet transmission within stripes being close to 1, and on the coalescence property in order to counteract the bias due to shortcomings with this assumption. In this section we investigate conformance of both of these assumptions to measurements of stripes transmitted across a number of end-to-end paths in the Internet. Although these experiments did not access the transmission properties of individual links (logistically very difficult to measure), they would be able to detect link-wise departures from the assumptions, since these would also be reflected in the properties of end-to-end paths over non-conformant links.

A. Measurement Infrastructure

We conducted the experiments using the National Internet Measurement Infrastructure (NIMI) [19]. NIMI consists of a number of measurement platforms deployed across the Internet (primarily in the U.S.) that can be used to perform end-to-end measurements. We made the measurements using the `zing` utility, which sends UDP packets in selectable patterns, recording the time of transmission and reception. We extended `zing` to transmit unicast stripes to multiple destinations, minimizing the spacing between packets in a stripe by precomputing the packets to send (including their MD5 integrity checksum, the most computationally expensive part of constructing a `zing` packet) and then transmitting them with back-to-back system calls, resulting in inter-packet spacings of about 40μsec.

A key point is that for our measurements we did *not* actually send packets to multiple destinations, because we had no way of calibrating true inference of internal loss characteristics, which would require measurement inside the network. Instead, the results we report are all for stripes sent to the *same* destination, with the goal being

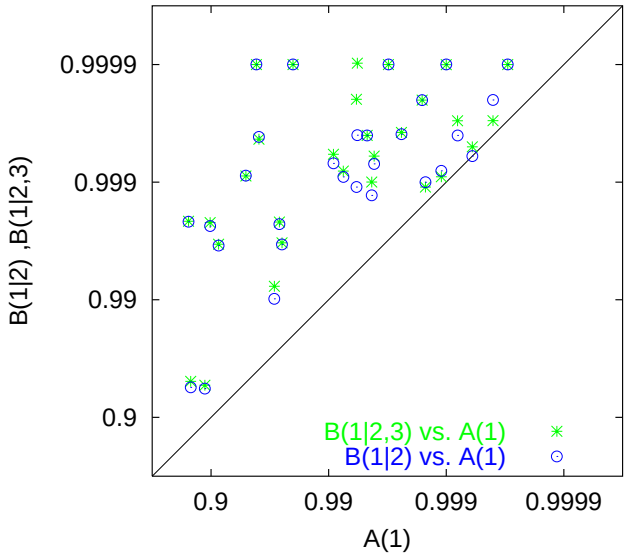


Fig. 2. SCATTER PLOT OF TRANSMISSION PROBABILITIES IN 28 NETWORK EXPERIMENTS. Conditional vs. marginal end-to-end transmission probabilities. Probabilities for 3-packet stripes mostly meet or exceed those for 2-packet stripes.

to assess the conditional loss probability and coalescence properties.

We gathered a total of 63 successful measurements between 35 NIMI sites, each measurement recording at both sender and receiver the transmission of either 100,000 flights of stripes of 3 packets, with separations exponentially distributed with a mean of 100 msec; 10,000 flights of stripes of 10 packets, separated by a mean of 300 msec (we also analyzed the first 3 packets in each stripe as another dataset of 3-packet stripes); or 20,000 flights of stripes of 3 packets, separated by a mean of 500 msec. All measurements were made at either 2PM EDT (a busy time) or 2AM EDT (a fairly unloaded time). There was no noticeable change in behavior as we varied the inter-stripe spacing from 100 msec to 500 msec.

Of the 63 traces, 7 exhibited no loss whatsoever, and consequently we had to eliminate them as they could not be used to study loss inference. Of the remaining 56, fully half (28) had conditional loss probabilities of 1, reflecting perfect loss correlation just as we would have if using multicast traffic instead of unicast. This finding is highly encouraging for the efficacy of unicast loss inference.

In the remainder of this section, we analyze the properties of the 28 traces that did not exhibit perfect correlation.

	$\widehat{B}(1\|2,\ldots,w)/\widehat{B}(1\|2,\ldots,w-1)$				
	$w=2$	$w=3$	$w=4$	$w=5$	$w=6$
min.	1.0000	1.0000	1.0000	1.0000	1.0000
mean	1.0189	1.0002	1.0000	1.0001	1.0001
max.	1.1812	1.0021	1.0003	1.0005	1.0003

TABLE I
COALESCENCE OF TRANSMISSION IN NETWORK EXPERIMENTS. RATIOS OF END-TO-END CONDITIONAL TRANSMISSION PROBABILITIES IN STRIPES OF WIDTH 2 TO 6. MINIMUM, MEAN AND MAXIMUM OF RATIOS OBSERVED IN 19 TRACES STRIPES OF WIDTH 10. MINIMUM RATIO 1 CONFORMS WITH COALESCENCE PROPERTY.

B. Transmission Probabilities

Marginal Probabilities. The packet loss rate varied between zero and about 14% over the experiments. The marginal packet loss rates for different positions in the stripe displayed some heterogeneity. The heterogeneity was most pronounced at the start of the stripe, with the loss rate for the second packet in a stripe being typically 1.19 times greater than that of the first. Moving further along the stripe, loss rates differed between successive positions typically by up to a typical factor of 1.03.

Conditional Probabilities. We can estimate the error involved in the stripe method by comparing conditional and marginal transmission probabilities within the stripe. A scatter plot of the conditional vs. marginal probabilities for 2 and 3 packet stripes in 28 experiments is shown in Figure 2. Higher points represent smaller relative error; conversely for points near the line the error is of the same order of magnitude as the marginal probability to be estimated. For both 2 and 3 packet stripes, the end-to-end conditional transmission probabilities $\widehat{B}$ are noticeably larger than the marginal transmission probabilities $\widehat{A}$, with those for the 3 packet stripe being at least as large as those for the 2 packet stripes in almost all cases. A conditional probability of 1 would signify perfect correlations. We can characterize this error arising from $\widehat{B} < 1$ through the ratio $(1-\widehat{B})/(1-\widehat{A})$ when $\widehat{A} \neq 1$. This represents the proportion of the reported loss rate which is typically in error due to imperfect correlations. For 2-packet stripes, the median value of this ratio was 0.12. (So, for example, an estimated loss rate of 1% would be in error by about 0.12%). The median ratio fell to to 0.09 for 3 packet stripes.

Coalescence We calculated end-to-end conditional transmission probabilities $\widehat{B}(1|2,3,\ldots w)$ for stripes of width w between 1 and 6. (When $w = 1$ this just denotes the marginal probability $\widehat{A}(1)$). A necessary condition for coalescence is that the ratios $\widehat{B}(1|2,\ldots,w)/\widehat{B}(1|2,\ldots,w-$

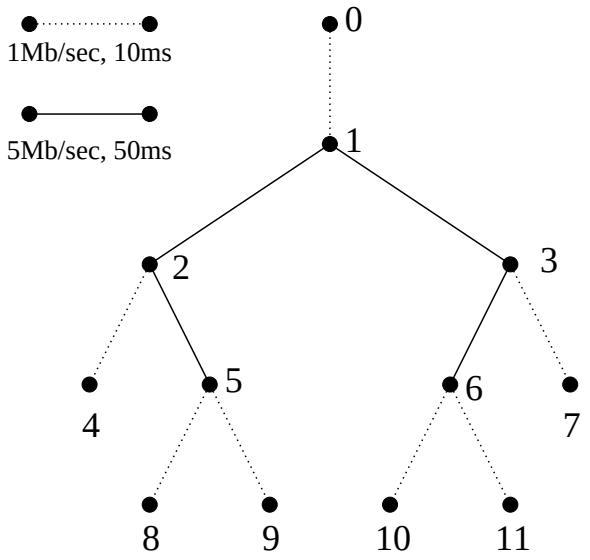


Fig. 3. SIMULATION TOPOLOGY

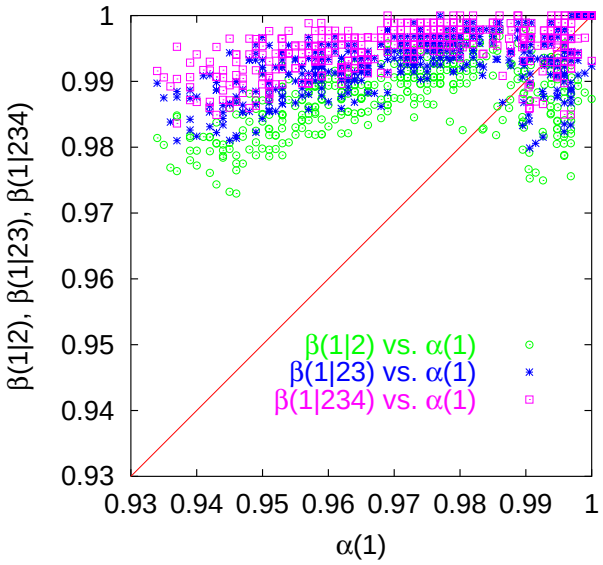


Fig. 4. CONDITIONAL TRANSMISSION PROBABILITIES IN SIMULATIONS. Scatter-plot of conditional vs. marginal link transmission probabilities for 2, 3 and 4 packet stripes. Conditional probabilities increase with stripe width.

1) be ≥ 1. We determined the ratios over 19 experiments with stripes of width 10. In only two instances were the ratios less than 1, and in these cases by a magnitude of only about 10^{-6}. This is a far smaller magnitude than that by which the ratio typically exceeds 1, as is seen from the statistics displayed in Table I: the minimum, mean, and maximum for each w over the 19 experiments. The ratios are largest for $w = 2$, falling off close to 1 as w increases beyond 3. This suggests that the additional bias correction obtained by increasing stripe width is almost negligible for stripes wider than 3 packets, at least under the network conditions and the range of loss probabilities exhibited in these traces.

C. Interpretation

The network experiments are encouraging for unicast-based inference. First, in half of the traces the stripes exhibited perfect correlations. If this property were reproduced in stripes to multiple destinations, their statistical properties would be identical to that of multicast traffic for the purposes of link loss inference. Second, in traces with imperfect correlations, the conditional transmission probabilities within the stripe were considerably higher than the marginal probabilities, slightly more for the 3 packet stripe than the 2 packet stripe. This indicates that the bias due to ignoring the imperfection in correlations is relatively small. Third, traces exhibited coalescence for the stripe widths considered, indicating that the bias can be compensated for by using wider stripes, although the incremental benefit grew smaller for larger stripe widths. These factors lead us to expect that striped unicast probing will be quite effective for loss inference under real network conditions.

IV. SIMULATION RESULTS

A. Methodology

The experiments of Section III give us confidence that the statistical properties of stripe transmission make stripes suitable as probes for inference. However, the experiments do not enable us to corroborate the accuracy of the estimators for real network traffic. Instead, we employ simulation to get a sense of how accurate the estimators might be in practice.

We used the `ns` simulation environment [17]; this enables the representation of transport-protocol detail of packet transmissions, with packet loss due to buffer overflows at nodes as stripes compete with background traffic. The simulations reported in this paper used the topology of Figure 3. The different link speeds and delays are intended to characterize low speed/low delay links at a network edge connected by high speed/high delay links in the network interior. The goal is to study the methodology in a simplified environment to look for major problems, not to make a definitive assessment of the methodology.

Background traffic comprised a mixture of sessions over TCP and exponential on-off sources. There were on average 11 sessions per link direction. The buffer on each link accommodated 20 packets. Measurement probes comprised stripes with a 1μsec interpacket time. Stripes were generated periodically with an inter-stripe of 16 msec. The tree was covered by cycling through thirty stripes $S(i, j)$

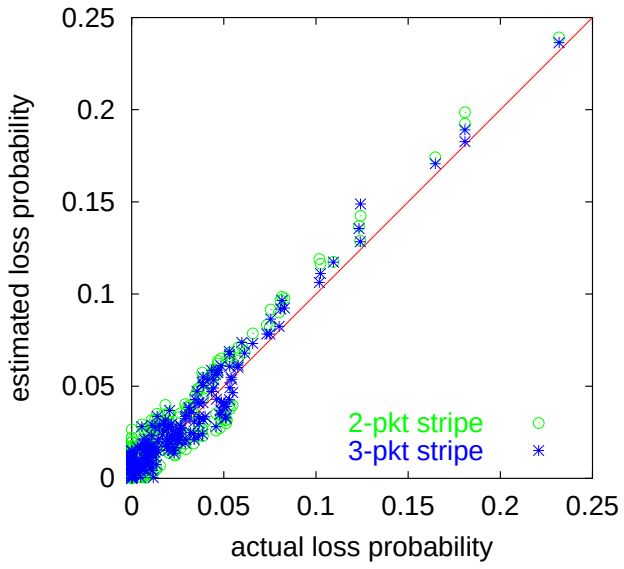


Fig. 5. Inferred vs. actual link loss rates in simulations. 3 packet and 2 packet substripes. Scatter plot for 100 experiments.

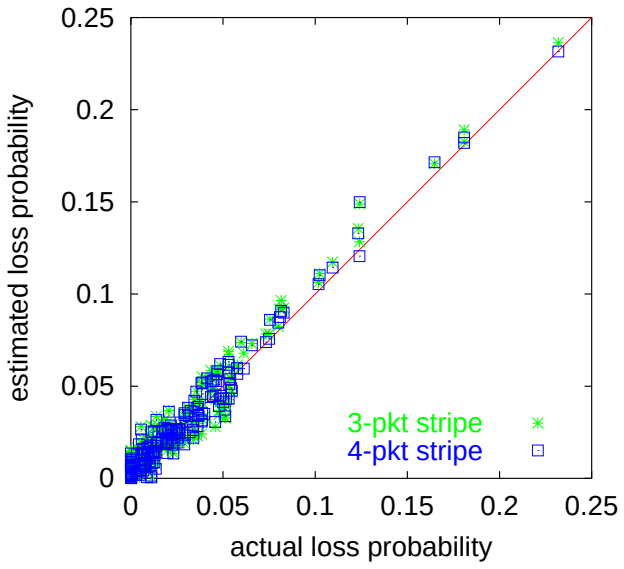


Fig. 6. Inferred vs. actual link loss rates in simulations. 4 packet stripes and 3 packet substripes. Scatter plot for 100 experiments.

over pairs of distinct receivers i, j. During an experiment, each stripe was transmitted 1,000 times. We conducted a set of 100 experiments using 4 packet stripes. To compare the estimator performance under the different stripe lengths we considered the 2 and 3-packet substripes obtained using the first two and three packets in each stripe. In order to evaluate the method, the inferred loss rates were compared with internal link loss rate as determined by instrumentation of the simulation. Link loss rates were computed considering only the first probe in the stripe.

B. Conditional and Marginal Transmission Probabilities

We first examine the statistical properties of the underlying link loss processes. Figure 4 is a scatter plot of conditional vs. marginal transmission probabilities for 2, 3 and 4 packet stripes. Observe that conditional probabilities increase with stripe width. We summarize the likely relative errors in each case though the statistics of the ratio $(1-\widehat{\beta})/(1-\widehat{\alpha})$ of conditional to marginal loss probabilities. For 2 packet stripes the median ratio was 0.32 (i.e., a relative error of 32%). The ratio fell to 0.20 for 3 packet stripes, and further to 0.12 for 4-packet stripes.

These errors are somewhat greater than those observed for end-to-end transmission in the network experiments. We believe this may be associated with a greater heterogeneity in marginal transmission rates that we observed in the simulations; loss rates grew by about 30% between successive positions for the first 4 packets of a stripe. Recall from Section III-B that in the network experiments, the largest such ratio was 19%, and typical ratios were 3%.

	stripe width		
	2	3	4
mean	0.0099	0.0075	0.0063
s.d.	0.0064	0.0057	0.0052

TABLE II
ESTIMATION ERROR IN SIMULATIONS AS FUNCTION OF STRIPE WIDTH. MEAN AND STANDARD DEVIATION OF ABSOLUTE DIFFERENCE BETWEEN INFERRED AND ACTUAL LOSS RATES. ERRORS ARE MINIMIZED FOR 4-PACKET STRIPES.

The stronger growth in loss ratios along the stripe in the simulations may be due to the larger size of the stripe relative to buffer size (20 packets) as compared with that in real networks.

C. Accuracy of Inference

Finally, we compare inferred and actual link loss rates in the simulations. We display scatter plots of inferred vs. actual loss for 2 and 3 packet stripes in Figure 5, and 3 and 4 packet stripes in Figure 6. The same number of stripes was used in each case. From the figures we observe that accuracy increases with wider packet stripes as exhibited by the clustering about the line $y = x$. In Table II we summarize the statistics of the absolute error, i.e., the modulus of the difference between the inferred and actual link loss rates. This is just under 1% in the worst case, i.e., for the 2 packet stripe, and 0.63% in the best case, i.e., the 4 packet stripe. Thus, by exploiting the coalescence property, we

ha ve achieved a 40% reduction in absolute error, by simply increasing the the stripe length from two to four.

V. Conclusions and Further Work

In this paper we have proposed a method of using end-to-end unicast probing to infer the loss characteristics of the network interior. The method relies on using collections of unicast probes, called stripes, dispatched back-to-back to different destinations, in order to mimic the effect of a notional multicast packet that followed the same path. We infer internal loss rates by applying an estimator developed for multicast inference to the unicast receiver traces. This estimator is unbiased when the transmissions of a stripe's probes on a given link are perfectly correlated. Imperfect correlations lead to bias, but we prove that this can be compensated for by using wider stripes, provided that the stripe transmissions obey a certain correlation property that we call coalescence. This is the property that successful transmission of a given packet in the stripe becomes more likely when more other packets from the stripe have been successfully transmitted.

Our network experiments show that for end-to-end transmission, correlations within stripes are very high, even perfect in some cases. Moreover, the coalescence property was found to hold in virtually all cases examined. Together these properties lead us to expect that inference from striped unicast probes will be effective in estimating link loss rates.

Our next step in network experimentation is to directly assess the method by performing corroborative measurements in the network interior. This entails taking measurements on paths over which probe traffic flows; then comparing actual loss rates with inferred loss rates on internal paths.

Currently, such corroboration is available to us only in simulation experiments. The `ns` simulations showed good agreement between inferred and actual loss rates; the typical error in these experiments was about 1% for the 2-packet stripe, falling to 0.63% when the stripe width was increased to 4.

Our next step in simulation will be to investigate the magnitude of these effects for systems with larger buffers and more diverse background traffic, which are more representative of actual networks.

In this paper we have concentrated on estimation of link probabilities for the first packet of a stripe. However, due to heterogeneity of loss along the stripe, such estimates may not be representative of typical packets, e.g., packets contained within a burst. Clearly, the present method could be extended, through use of other stripe templates, to estimate link probabilities for packet in positions other than the first. In the future we hope to increase the accuracy of inference by tuning the stripe properties to the burst structure observed in background traffic.

Finally, we remark that a number of other multicast-based estimators–namely those for delay distributions [15], for delay variances [7], and logical multicast topology [3]–have the potential to be adapted in the same manner as was done for loss estimators in this paper. We feel that our promising results on unicast-based loss estimation warrant extending the estimator to these other settings.

Acknowledgement

We thank Ramon Caceres for his help with `ns`. Many thanks to Andrew Adams, his NIMI colleagues Matt Mathis and Jamshid Mahdavi, and the many NIMI volunteers who host NIMI measurement servers, for facilitating our Internet measurements.

References

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior, IEEE Communications Magazine., May 2000.

[2] R. Caceres, N.G. Duffield, J.Horowitz, D. Towsley, "Multicast-Based Inference of Network Internal Loss Characteristics", IEEE Trans. on Information Theory, 45: 2462–2480, 1999.

[3] R. Caceres, N.G. Duffield, J.Horowitz F. Lo Presti, D. Towsley, "Statistical Inference of Multicast Network Topology", in Proc. IEEE Conf. on Decision and Control, Phoenix, AZ, Dec. 1999.

[4] R. Carter, M. Crovella, 'Measuring bottleneck link-speed in packet-switched networks,' *Performance Evaluation*, 27&28, 1996.

[5] M. Coates, R. Nowak. "Network loss inference using unicast end-to-end measurement, to appear, *Proc. ITC Conf. IP Traffic, Modeling and Management*, Sept. 2000.

[6] A.B. Downey. "Using pathchar to estimate Internet link characteristics," *Proc. SIGCOMM'99* Sept. 1999.

[7] N.G. Duffield and F. Lo Presti, "Multicast Inference of Packet Delay Variance at Interior Network Links", in Proc. IEEE Infocom 2000, Tel Aviv, March 2000.

[8] S. Floyd. "Simulator tests." July 1995; revised May 1997. See http://www.aciri.org/floyd/papers/simtests.ps.Z

[9] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, 1(4), August 1993.

[10] V. Jacobson, "Congestion Avoidance and Control," *Proc. SIGCOMM '88*, pp. 314-329, Aug. 1988.

[11] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see ftp://ftp.ee.lbl.gov/pathchar

[12] S. Keshav. " A control-theoretic approach to flow control," *Proc. SIGCOMM'91*, 3–15, Sept. 1991.

[13] K. Lai, M. Baker,"Measuring link bandwidths using a deterministic model of packet delay," *Proc. SIGCOMM 2000*, to appear.

[14] B.N. Levine, S. Paul, J.J. Garcia-Luna-Aceves, "Organizing multicast receivers deterministically according to packet-loss correlation", Preprint, University of California, Santa Cruz.

[15] F. Lo Presti, N.G. Duffield, J.Horowitz and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions", submitted for publication, September 1999.

[16] `mtrace` – Print multicast path from a source to a receiver. See ftp://ftp.parc.xerox.com/pub/net-research/ipmulti

[17] `ns` – Network Simulator. See http://www-mash.cs.berkeley.edu/ns/ns.html

[18] V. Paxson. "End-to-End Internet Packet Dynamics," *Proc. ACM SIGCOMM '97*, September 1997.

[19] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," IEEE Communications Magazine, Vol. 36, No. 8, pp. 48–54, August 1998.

Tree Layout for Internal Network Characterizations in Multicast Networks *

Micah Adler, Tian Bu, Ramesh K. Sitaraman, Don Towsley

Technical Report 2000-44
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
July, 2000
micah,tbu,ramesh,towsley@cs.umass.edu

Abstract

There has been considerable activity recently to develop monitoring and debugging tools for a multicast session (tree). With these tools in mind, we focus on the problem of how to lay out multicast sessions so as to cover a set of links of interest within a network. We define three variations of this layout (cover) problem that differ in what it means for a link to be covered. We then focus on the identifiability problem, to determine whether a given set of candidate multicast trees can cover the set of links of interest; and the minimum cost problem, to determine the minimum cost set of trees that cover the links in question. We establish efficient algorithms to solve the identifiability problem and show that, with few exceptions, the minimum cost problems are NP-hard and that even finding an approximation within a certain factor is NP-hard. One exception is when the underlying network topology is a tree. For this case, we demonstrate an efficient algorithm that finds the optimal solution. We also present several computationally efficient heuristics and their evaluation through simulation. We find that two heuristics, a greedy heuristic that combines sets of trees with three or fewer receivers, and a heuristic based on generalizing our tree algorithm, both perform reasonably well. The remainder of the paper applies our techniques to the vBNS and Abilene networks, examining the effectiveness of the different heuristics and the sensitivity of the costs to the choice of routing algorithm.

1 Introduction

Multicast is a technology that shows great promise for providing the efficient delivery of content from a single source to many receivers. An interoperable networking infrastructure is nearly in place (PIM-SM/MSDP/MBGP,SSM) and the development of mechanisms for congestion control and reliable data delivery are well under way [3, 7]. However, deployment of multicast applications lags behind, in large part because of a lack of debugging and monitoring tools. Recently, several promising approaches and protocols have been proposed for the purpose of aiding the network manager or the multicast application designer in this task. These include the use of end-to-end measurements for inferring internal behavior on a multicast

*This work is sponsored in part by the DARPA and Air Force Research Laboratory under agreement F30602-98-2-0238.

tree [1], the development of the multicast route monitor (MRM) protocol [2], and a number of promising fault monitoring tools, [10, 13]. All of these address the problem of identifying performance and/or fault behavior on a *single multicast tree.*

Although considerable progress has been made in developing tools for a single tree, little attention has been paid on how to apply these tools to monitor an entire network, or even a subset of the network. We address this problem; namely, given a set of links whose behavior is of interest, how does one choose a set of minimum cost multicast trees within the network on which to apply these tools so as to determine the behavior of the links in question? Resolution of this problem is especially important as poorly designed sets of measurements can easily overwhelm network resources. The choice of trees, of course, is determined by the multicast routing algorithm. This raises a related question, namely, does the multicast routing algorithm even allow a set of trees that that will allow one to determine the behavior of the links of interest. We refer to this latter problem as the Multicast Tree Identifiability Problem (MTIP) and the first problem as the Minimum cost Multicast Tree Cover Problem (MMTCP).

We refer to the behavior measurement of a link as a *link measure*. Note that solutions to MTIP and MMTCP depend on details of the mechanism used to determine the link measures. Consequently, we introduce three versions of these problems, the weak, strong, and medium cover problems. Briefly, the weak cover problem is based on the assumption that it is sufficient that each link of interest appear in at least one tree. The strong cover problem requires that each link occur between two branching points in at least one tree. The medium cover problem relaxes this last requirement and instead requires that the set of trees covering the link provide enough information to determine the link measure of interest.

Briefly, the paper makes the following contributions.

- We establish efficient algorithms for solving the identifiability problems. These are sufficiently general to apply to unicast-based end-to-end measurements as well, e.g.,[11].
- We establish that the cover problems are NP-hard and that in some cases, finding an approximation within a certain factor of optimal is also NP-hard. Thus, we also propose several heuristics and show through simulation that a greedy heuristic that iteratively combines trees containing a small number of receivers performs reasonably well.
- We provide polynomial time algorithms that find optimal solutions for a restricted class of network topologies, including trees. This algorithm can be used to provide a heuristic for sparse, tree like networks. This heuristic is also shown through simulation to perform well.
- We apply our techniques to the vBNS and Abilene networks, examining the effectiveness of the different heuristics and the sensitivity of the costs to the choice of routing algorithm.

To motivate the need for different cover problems, let us focus on the MINC approach for inferring link-level loss behavior [1]. Given a sender and two or more receivers, there exist unbiased estimates of the packet loss probabilities on each segment of the tree, whether it lies between source and branch point, branch points, or branch point and receiver. Consider the simple topology illustrated in Figure 1(a). Assume that only the two trees illustrated in Figure 1(b) and (c) are available to cover links. Suppose that we are interested in the loss behavior of the set of links $\{(2,4),(4,6),(4,7)\}$. In this case, applying the MINC technique to the tree in 1(b) will produce estimates for the loss rates for those links. This is an example where the tree provides a strong cover for the links of interest. Suppose that we are now interested in link $(3,4)$. Neither tree provides a strong cover for this link. However, the tree in 1(b) allows us to determine the loss rate for link $(4,6)$ and the tree in 1(c) allows us to determine the loss rate of path $(3,6)$. If we assume that loss events are independent between links, then these loss rates can be used to compute the loss rate for link $(3,4)$. Thus, the two trees provide a medium cover of the set of links $\{(2,4),(4,6),(4,7),(3,4)\}$.

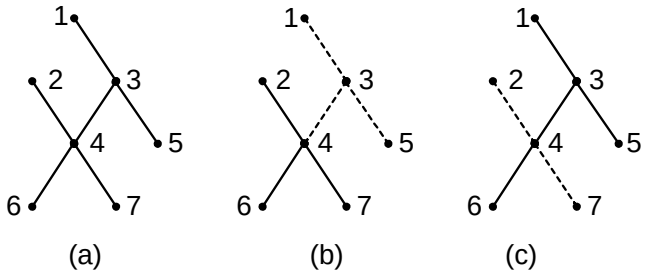


Figure 1: A simple topology (a) with two possible trees: (b) where 2 sends to 6 and 7, and (c) where 1 sends to 5 and 6.

A weak cover may suffice to provide a warning that some link is causing a problem. This can trigger more bandwidth and computation intensive diagnostics to determine what link is causing the problem.

A number of end-to-end measurement techniques have been developed for network tomography. In addition to the MINC approach which applies to multicast, there are several promising unicast-based techniques. For example, [5, 8] attempt to emulate multicast through the use of packet pairs so that multicast-based analysis techniques can be applied. An algebraic approach has been proposed in [11] which can be used to infer link round trip times based on path round trip time measurements. However, none of these deal with the problem of designing measurements so as to minimize the overhead imposed on the network. We believe that our techniques can be adapted to be used with these measurement methodologies.

The remainder of the paper proceeds as follows. Section 2 presents the model for MTIP and MMTCP, as well as the three types of covers we consider. Section 3 describes the efficient algorithms for solving MTIP. Section 4 introduces several approximation algorithms and heuristics for MMTCP. In Section 5 we present efficient algorithms that find the optimal MMTCP solution for the special case where the underlying

network topology is a tree. Section 6 presents the results of simulation experiments on the VBNS and Abilene networks as well as randomly generated networks. Last, Section 7 concludes the paper.

2 Model and Assumptions

We represent a network N by a directed graph $N = (V(N), E(N))$ where $V(N)$ and $E(N)$ denote the set of nodes and links within N respectively. When unambiguous, we will omit the argument N. Our interest is in multicast trees embedded within N. Let $S \subseteq V(N)$ be a set of possible multicast senders, and let $R \subseteq V(N)$ be a set of possible multicast receivers. Let $T = (V(T), E(T))$ denote a directed (multicast) tree with a source $s(T)$ and a set of leaves $r(T)$. We require that $s(T) \in S$ and $r(T) \subseteq R$. Let A be a mapping that takes a source $s \in S$ and receiver set $r \subseteq R$ and returns a tree $A(s, r)$. In the context of a network, A corresponds to the multicast routing algorithm. Examples include DVMRP [9], and PIM-DM and PIM-SM [6]. Let $\mathcal{T}(A, S, R) = \{A(s, r) : s \in S, r \subseteq R/\{s\}\}$, i.e., $\mathcal{T}(A, S, R)$ is the set of all possible multicast trees that can be embedded in N using multicast routing algorithm A. We shall henceforth denote $\mathcal{T}(A, S, R)$ by $\mathcal{T}(S, R)$, omitting the dependence on A.

We now associate a cost with a multicast tree $T \in \mathcal{T}(S, R)$. We assume that it can be expressed as

$$C(T) = C_T^0 + \sum_{l \in E(T)} C_l, \tag{1}$$

where the first term can be thought of as a "per tree cost" and the second is a "per link cost". The two problems of interest to us are as follows:

Multicast Tree Identifiability Problem. Given a set of multicast trees $\Psi \subseteq \mathcal{T}(S, R)$, and a set of links $L \subseteq E$, is L identifiable by the set of trees Ψ?

Minimum cost Multicast Tree Cover Problem. Given $S, R \subseteq V$ and $L \subseteq E$, what is the minimum cost subset of $\mathcal{T}(S, R)$ sufficient to cover L? In other words, find $\Psi \subseteq \mathcal{T}(S, R)$ that covers L and minimizes

$$C(\Psi) = \sum_{T \in \Psi} C(T)$$

We distinguish three types of solutions to both of these problems. These solutions differ in what exactly is meant by a cover. We say that a node v is a branch point in tree T if v is either a root or a leaf, or v has more than one child. A path $l = (v_1, v_2, \ldots, v_n)$ is said to be a *logical link* within T if v_1 and v_n are branch points, $v_2, \ldots, v_{n-1}$ are not, and$(v_i, v_{i+1}) \in E(T)$, $i = 1, \ldots, n-1$.

- **Strong cover:** Given a set of trees Ψ, Ψ is the strong cover of link $l = (u, v)$ if there exists a $T \in \Psi$ such that both u and v are branch points in T. Ψ is the strong cover of a link set L if $\forall l \in L$, Ψ is the strong cover of l.

- **Medium cover:** Given a set of trees Ψ, let L^* be the set of all paths for which Ψ is the strong cover, and let M^* be a set of observed link measures for each $l^* \in L^*$. We say that Ψ is the medium cover of link l if any observed M^*, resulting from a situation where successful transmission over each path is possible, uniquely determines the link measure for l. We say that Ψ is the medium cover of a link set L if $\forall l \in L$, Ψ is the medium cover for l.

- **Weak cover:** Given a set of trees Ψ, Ψ is the weak cover of link l if there exists a $T \in \Psi$ such that $l \in E(T)$. We say that Ψ is the weak cover of a link set L if $\forall l \in L$, Ψ is the weak cover of l.

We refer to the problems of finding these types of solutions as *S-MTIP/S-MMTCP, W-MTIP/W-MMTCP* and *M-MTIP/M-MMTCP* respectively. Several cases are of interest to us. One is where $L = E$, i.e., where the objective is to cover the entire network. A second is where L consists of one link, $|L| = 1$. If, $\forall l$, we set $C(l) = 0$, the problem becomes that of covering the link set L with the set of trees with minimum total per tree cost.

3 The Multicast Tree Identifiability Problem

In this section, we consider the identifiability problem associated with the weak, strong, and medium cover problems. In the case of weak covers, the identifiability problem is straightforward. It suffices to check whether each link in L appears in one of the trees within Ψ. The identifiability problem is also straightforward in the case of strong covers. In this case, L is identifiable iff each link within L lies between branch points in at least one tree within Ψ.

The identifiability problem for medium covers is more interesting and has application to a much larger set of network characterization problems than the one being considered in this paper, e.g., [11]. Unlike the weak and strong cover problems, we have results only for summable link measures. Fortunately, this is not restrictive as such measures include loss probabilities, average delays, delay variances, and others.

We say that a link measure is *summable* if, for a path p consisting of the set of links P,

$$\beta_p = \sum_{l \in P} m_l, \tag{2}$$

where β_p and m_l are the link measures for the path p and the link l, respectively. We also require that m_l be finite if it is possible to transmit across link l. Expected link delay is an example of a summable link measure.[1] The logarithm of the probability of successful transmission is summable provided that loss events are independent from link to link. Note that if it is possible to transmit on a link, then this link

[1] Note that for link delay, "possible to transmit" means the expected delay is finite.

measure is finite. Given the logarithm of the success probability, we can of course compute the loss or success probability.

We focus now on the identifiability problem associated with the medium problem. Let $T \in \Psi$. Define a segment in T to be a path between either the root of T and the closest branch point, two neighboring branch points, or a branch point and a leaf of T. Let S be the set of all segments within the trees contained in Ψ. Each segment $p \in S$ is a path within Ψ. The fundamental assumption underlying our work is that Ψ allows us to obtain β_p, for all $p \in S$, i.e., S is identifiable from Ψ. Note also that S corresponds to a set of logical links that corresponds to the strong cover of Ψ. Now, (2) holds for all $p \in S$. If we introduce the $|S| \times |E|$ matrix B where $B_{i,j} = 1$ if link j belongs to segment i and 0 otherwise, then we have the following matrix equation

$$Bm = \beta \tag{3}$$

where the components of m are m_k and the components of β are β_k. We state without proof the following identifiability result.

Theorem 1 *Let Ψ be a set of multicast trees. Provided that the link measures are summable, Ψ identifies the link measures m_l, $l \in L$ iff for β such that the components correspond to the number of links in the segment, there is a unique set $\{m_l, l \in L\}$ that satisfies equation (3).*

Determination of whether L is identifiable or not can be efficiently performed. Choose β_k to be the number of links in segment $k \in S$. Apply Gaussian elimination to B to obtain its reduced row echelon form matrix B'. L is identifiable if $\forall l \in L$, there is some row in B' where the column corresponding to link l is the only non-zero value. This corresponds to the existence of a unique set $\{m_l, l \in L\}$ satisfying (3).

4 Approximating the Minimum cost Multicast Tree Cover Problem

We have defined three types of Multicast Tree Cover Problems: the S-MMTCP, the W-MMTCP, and the M-MMTCP. Unfortunately, as the following theorem shows, not only are these problems NP-Complete, we cannot even expect to find even a good quality approximation to these problems. In particular, we can demonstrate the following (the proof is deferred to the full version of the paper):

Theorem 2 *For each of S-MMTCP, W-MMTCP, and M-MMTCP, it is NP-Hard to find a solution that is within a factor of $(1-\epsilon)\ln|L|$ of the optimal solution, for any $\epsilon > 0$. These problems are also still NP-Hard even with the restriction that $\forall l, C(l) = 0$.*

Since we cannot expect to solve these problems exactly, in the remainder of this section, we focus on approximation algorithms and heuristics for good solutions. In Section 4.1, we focus on approximation

algorithms for the case where the goal is to minimize the total cost of setting up the multicast trees. We provide polynomial time algorithms that have a provable bound on how close to optimal the resulting solution is for the S-MMTCP and the W-MMTCP. In Section 4.2, we describe extensions to these algorithms for the problem of approximating the general MMTCP. The resulting algorithms only run in polynomial time when the number of possible receivers is $O(\log |E|)$, and thus, to approximate the general MMTCP more efficiently, we also propose a heuristic that always finds a solution to the general MMTCP in polynomial time. In Section 6, we experimentally verify the quality of the solutions found by this heuristic.

4.1 Minimizing the total per-tree cost

We first study how to approximate the MMTCP when the goal is only to minimize the total cost for setting up the multicast trees but not the cost for multicast traffic to travel links, i.e., $C_l = 0$ in (1). This problem is simpler, since without a cost for link traffic, if a sender is performing a multicast, there is no additional cost for sending to every receiver. Thus, we can assume that any active sender multicasts to every possible receiver. Note, however, that by Theorem 2, even this special case is NP-Hard to approximate within better than a $\ln |L|$ factor. We describe algorithms for this problem that achieve exactly this approximation ratio. These algorithms rely on the fact that when $C(l) = 0, \forall l$, then both the W-MMTCP and the S-MMTCP can be solved using algorithms for the weighted Set-Cover problem, which is defined as follows:

- The **weighted Set-Cover** problem: given a finite set $B = \{e_1, e_2, ..., e_m\}$ and a collection of subsets of B, $\mathcal{F} = \{B_1, B_2, ..., B_n\}$, where each $B_i \subseteq B$ is associated with a weight w_i, find the minimum total weight subset $\hat{\mathcal{F}} = \{\hat{B_1}, ..., \hat{B_k}\} \subseteq \mathcal{F}$ such that each $e_i \in B$ is contained in some $\hat{B_j} \in \hat{\mathcal{F}}$.

To use algorithms for the weighted Set-Cover problem to solve the S-MMTCP or W-MMTCP, we simply set $B = L$, $B_i = \{l \in L$ such that l is in the cover (strong or weak, respectively) produced by T_i, where T_i is the tree produced by sender i multicasting to every receiver$\}$. The weight of B_i is the per-tree cost of multicasting from sender i. Any solution to the resulting instance of the weighted Set-Cover problem produces a S-MMTCP (W-MMTCP, resp.) solution of the same cost.[2] Using this idea, we introduce two algorithms for the MMTCP: a greedy algorithm modeled after a weighted Set-Cover algorithm analyzed by Chvatal [4], and an algorithm that uses 0-1 integer programming, constructed using a weighted Set-Cover algorithm analyzed by Srinivasan [12].

Greedy algorithm: The intuition behind the greedy algorithm is simple. Assume first that the per-tree cost is the same for every multicast tree. In this case, the algorithm, at every step, chooses the multicast tree that covers the most remaining uncovered links. This is repeated until the entire set of links is covered. When

[2]Note that this technique does not work for the case of a medium cover, since the set of links in the medium cover of a tree T_i will depend on what other trees are also being used.

different trees have a different per-tree cost, then instead of maximizing the number of new links covered, the algorithm maximizes the number of new links covered, divided by the cost of the tree. Intuitively, this maximizes the "profit per unit cost" of the new tree that is added.

The details of the algorithm are shown in Figure 2. This algorithm is easily seen to run in polynomial time for all three types of covers. For the W-MMTCP and S-MMTCP , Theorem 3 provides a bound on how good an approximation the algorithm produces. This algorithm can also be used to solve the M-MMTCP . We verify the quality of the approximation obtained for the M-MMTCP experimentally.

1. Apply the multicast routing protocol A to compute Ψ, the set of all multicast trees from a source in S to every receiver in R. $\forall T_i \in \Psi$, set its cost $c_i = C^0(T_i)$.
2. Set $J = \Psi, \hat{J} = \emptyset, \hat{L} = \emptyset$.
3. If $L = \hat{L}$, then stop and output $\hat{J}$.
4. $\forall T_i \in J \backslash \hat{J}$, set $L_i = Cov(\hat{J} \cup \{T_i\}, COVER)$. Find $T_i \in J \backslash \hat{J}$ that maximizes $|(L_i \cap L) \backslash \hat{L}|/c_i$.
5. $\hat{L} = \hat{L} \cup (L_i \cap L)$, $\hat{J} = \hat{J} \cup \{T_i\}$. Go to step 3.

Figure 2: The greedy algorithm to approximate MMTCP.

Theorem 3 *For any instance I of S-MMTCP or W-MMTCP with $C(l) = 0, \forall l$, the greedy algorithm finds a solution of cost at most $\left(\ln d + \frac{5}{8} + \frac{1}{2d}\right) \cdot OPT$, where $d = \min\left(|L|, \max_{i=1}^{n} |L_i|\right)$, and OPT is the cost of the optimal solution to I.*

We see from Theorems 2 and 3 that the performance of the greedy algorithm is the best that we can hope to achieve. However, these theorems only apply to the worst case performance; for the average case, the performance may be much better, and the best algorithm may be something completely different. We investigate this issue further by introducing a second approximation algorithm, based on 0-1 integer programming. We shall see in Section 6 that the 0-1 integer programming algorithm performs better than the greedy algorithm in some cases. The details of this algorithm are omitted from this version of the paper; we here only state the following theorem that demonstrates how good a solution is provided by this approach.

Theorem 4 *For any instance I of either the S-MMTCP or the W-MMTCP, the 0-1 linear programming algorithm finds a solution of cost at most $OPT(1 + O(\max\{\ln(m/OPT), \sqrt{\ln(m/OPT)}\}))$, where OPT is the optimal solution to I.*

We also point out that in the Set-Cover problem, if we let $d = \max |B_i|$, then even for $d = 3$, the set cover problem is still NP-hard. However, it can be solved in polynomial time provided that $d = 1$ or $d = 2$. Since we can transform the S-MMTCP and the W-MMTCP to Set-Cover problems, we know that the S-MMTCP and the W-MMTCP can be solved in polynomial time given that $\max_i |L \cap E(T_i)| \leq 2$ where T_i is the tree produced by sender i multicasting to every receiver.

4.2 Minimizing the total cost

We next look at the general MMTCP, where the goal is to minimize the total cost. In addition to the per-tree cost of the multicast trees used in the cover, this includes the cost of multicast traffic traveling on the links used by the trees. Both the greedy algorithm and 0-1 integer programming algorithm can be extended to approximate the general problem. For the greedy algorithm, we simply replace the first step with:

1. Apply the multicast routing protocol A to compute Ψ, the set of all multicast trees from a sauce in S to any subset of R. $\forall T_i \in \Psi$, compute its cost $c_i = C(T_i)$.

The bound from Theorem 3 on the quality of approximation achieved by the greedy algorithm also applies to this more general algorithm. However, the greedy algorithm has a running time that is polynomial in $|\Psi|$. For the algorithm of Section 4.1, $|\Psi| = |S|$, which results in a polynomial time algorithm, but for the more general algorithm considered here, $|\Psi| = |S| \cdot (2^{|R|} - 1)$. Thus, the more general approximation algorithm only has a polynomial running time when $|R| = O(\log n)$, where n is the size of the input to the MMTCP problem (i.e., the description of N, S, R and L). The analogous facts also apply to the 0-1 integer programming algorithms.

In order to cope with large values of $|R|$ in the general MMTCP, we also introduce the fast greedy heuristic, which always runs in polynomial time. Fast greedy is like the greedy algorithm, except that instead of considering all possible multicast trees (i.e., every tree from a sender to a subset of the receivers), it restricts itself to only those trees that contain 3 receivers (or, in the case of a weak cover, 1 receiver). There will be at most a polynomial number of such trees. Fast greedy then uses the greedy strategy to choose a subset of these trees covering all required links, and then merges the trees with the same sender. The details of this heuristic are described in Figure 3. We shall see in Section 6 that the performance of the fast greedy heuristic is often close to that of the greedy algorithm.

5 Finding the optimal solution in tree topologies

We saw in Theorem 2 that we cannot hope to find an efficient algorithm that solves any of version of the MMTCP in general. However, this does not rule out the possibility that it is possible to solve these problems efficiently on certain classes of network topologies. In this section, we study the MMTCP in the case that the underlying network topology N is a tree. This is motivated by the hierarchical structure of real networks, which can be thought of as having a tree topology with a small number of extra edges. We shall see in Section 6 that algorithms for the tree topology can be adapted to provide a very effective heuristic for such hierarchical topologies. We use this heuristic to provide good solutions to the W-MMTCP problem for the topologies of the vBNS network, as well as the Abilene network.

1. If COVER = 'strong' or 'medium', apply the multicast routing protocol A to compute Ψ, the set of all multicast trees that have one sender in S and three receivers in R.
If COVER = 'weak', apply the multicast routing protocol A to compute Ψ, the set of all multicast trees (paths) that have one sender in S and one receiver in R.
$\forall T_i \in \Psi$, compute its cost $c_i = C(T_i)$.
2. For all $T_i \in \Psi$, set $E_i = E(T_i)$.
3. Set $J = \Psi, \hat{J} = \emptyset, \hat{L} = \emptyset$.
4. If $L = \hat{L}$, then aggregate all the trees in $\hat{J}$ who share the same source node and output $\hat{J}$. Stop.
5. For all $T_i \in J \setminus \hat{J}$, set $\Psi_i = \hat{J} \cup \{T_i\}$ and aggregate all the trees in Ψ_i sharing a common source as one tree. Set $L_i = Cov(\Psi_i, COVER)$.
Find $T_i \in J \setminus \hat{J}$ that maximizes $|(L_i \cap L) \setminus \hat{L}|/c_i$.
6. $\hat{L} = \hat{L} \cup (L_i \cap L), \hat{J} = \hat{J} \cup \{T_i\}$. For each $T_j \in J \setminus \hat{J}$, if T_j shares the same source with T_i, $E_j = E_j \setminus E_i$, $c_j = \sum_{l \in E_j} C(l)$. Go to step 4.

Figure 3: Fast greedy heuristic to approximate MMTCP.

Our algorithm for the tree topology is guaranteed to find the optimal solution in polynomial time. We shall describe this algorithm for the (easier) case of the W-MMTCP. In order to describe this algorithm more concisely, we shall make some simplifying assumptions. In particular, we assume that N is a rooted binary tree, that the per tree cost of every multicast tree is zero, that $C((a,b)) = C((b,a))$ and that the cover requirement on a link can be satisfied from either direction. For the W-MMTCP, these assumptions can be removed by making the algorithm slightly more complicated, but without significantly increasing the running time of the algorithm. For the S-MMTCP, all of the assumptions can be removed except for the assumption that N is a binary tree.

The algorithm, based dynamic programming, starts by creating a table, with one row in the table for each link l of the tree, and $|S|^2$ entries in each row, labeled $C_{<l,i,j>}$, for $0 \leq i, j \leq |S|$. For link l connecting nodes $u, v \in L$, where u is closer to the root, let ST_l be the subtree rooted at node v, together with link l and node u. The value computed for entry $C_{<l,i,j>}$ is the minimum possible total cost for the tree ST_l (removed from the rest of the network), subject to the following conditions: all of the links that are required to be covered in N are covered in ST_l, u is a source that generates j multicast sessions that are routed across l, and u is also a receiver that receives i multicast sessions. If there are less than i senders in $ST_l - u$, or $j > 0$ and there are no receivers in $ST_l - u$, then we call the pair (i,j) *invalid* for link l, and the value of entry $C_{<l,i,j>}$ is set to infinity.

We compute the values in the table one row at a time, in decreasing order of the distance of the corresponding links from the root. When l is connected to a leaf of the tree N, it is straightforward to compute $C_{<l,i,j>}$ for all i, j, since if (i,j) is valid, then $C_{<l,i,j>} = (i+j)C_l$. We now show how to compute the remaining entries $C_{<l,i,j>}$ for a link l connecting nodes u and v, where u is closer to the root, and v is connected to two links m and n, as depicted in Figure 4. Since m and n are further from the root than l, we

can assume that $C_{<m,i_m,j_m>}$ and $C_{<n,i_n,j_n>}$ have already been computed, for $0 \leq i_m, j_m, i_n, j_n \leq |S|$.

We see that $C_{<l,i,j>} = \min(C_{<m,i_m,j_m>} + C_{<n,i_n,j_n>} + (i+j) * C(l))$, where the minimum is taken over all i_m, j_m, i_n, j_n that provide valid multicast flows through the node v. Which values of the flows are valid is checked using an algorithm described in Figure 14. For space reasons, this algorithm has been deferred to the appendix. Along with the value $C_{<l,i,j>}$, we store the values of the i_m, j_m, i_n and j_n that resulted in the minimum $C_{<l,i,j>}$. Call these values the *optimal indices* for $C_{<l,i,j>}$. If (i,j) is invalid for link l, $C_{<l,i,j>}$ is set to infinity. Also, if link l is in the to be covered set of links, $C_{<l,0,0>}$ is set to ∞. By proceeding in this fashion from the leaves to the root, we see that we can fill in the entire table.

To complete the algorithm, we attach a virtual link x to the root of N with $C(x) = 0$, and use the same technique to compute $C_{<x,0,0>}$. The minimum cost for covering the given set of required links in N is $C_{<x,0,0>}$. In order to find the actual multicast trees, we first follow the stored optimal indices from $C_{<x,0,0>}$ to the leaves of the tree to determine the actual optimal number of flows in either direction on each link of the tree. Given this information, a simple greedy algorithm finds a set of multicast trees that results in this number of flows. The description of this greedy algorithm is left to the full version of the paper. We present the details of our algorithm, which we call **Tree-Optimal**, in Figure 5.

Theorem 5 *The algorithm* **Tree-Optimal** *finds the optimal cost of a solution to the W-MMTCP in any binary tree in* $O(|E||S|^6)$ *steps.*

The proof is provided in the appendix. Following a similar idea, we can also construct an algorithm for solving the S-MMTCP. However, the strong cover requirements make that algorithm somewhat more complicated than the algorithm presented here.

In order to deal with the case that N is a tree of arbitrary degree (instead of just a binary tree), we transform N into a binary tree N'. To do so, choose any node to be the root of the tree N. Then, given a node r with n children, create a virtual node r', make it the child of node r and assign zero cost to the virtual link $\langle r, r' \rangle$. Then pick any $n-1$ other links attached to node r, and move them to node r'. We repeat this process until all nodes are binary; the resulting tree is N'. Since the cost of traveling any virtual link is zero, and no virtual link is in the set of to be covered links, the cost of an optimal solution for the topology N is the same as the optimal cost for the topology N'. Thus, we can find the optimal solution for N by transforming N into the topology N', finding an optimal solution for the topology N', and then transforming the solution for N' into an equal cost solution for N.

We also can deal with the general problem where each multicast tree has a per tree cost. We do this by adding a virtual node u and virtual link $\langle u, v \rangle$ for each source candidate v, and replacing the source candidate v with the virtual node u. We then assign the cost for initializing the multicast tree that was rooted at node v as the cost of the link $\langle u, v \rangle$. The problem then becomes that of finding the optimal cost covering without

per tree costs in the resulting network. If we also want to allow the cost of initializing different multicast trees at the same source to vary, we can attach a different virtual node u for each multicast tree rooted at v.

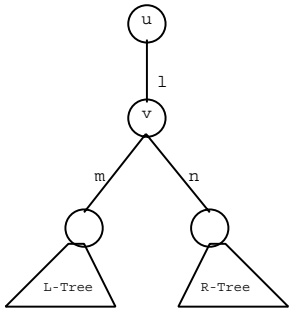


Figure 4: A link l incident to the same node as links m and n.

1. Add a virtual link x whose downstream node is the root of the tree and assign zero cost for traveling link x.
2. Create a table C, with each row labeled with a link l, and each column labeled by $<i,j>$, for $0 \leq i,j \leq |S|$. Initialize every entry in the table to $+\infty$.
3. For each link l, let SU_l and SD_l be the number of sources above and below l respectively. Let RU_l and RD_l be the number of receivers above and below l respectively.
4. $\forall l_i$, such that l_i is a link attached to a leaf node:
   if $(SD_{l_i} == 1)$ then $C_{<l_i,0,1>} = C(l_i)$
      if $(RD_{l_i} == 1)$ then $C_{<l_i,j,k>} = C(L_i) * (j+k), \forall j,k, 0 \leq j \leq SU_i, 0 \leq k \leq 1$. endif
   else
      if $(RD_{l_i} == 1)$ then $C_{<l_i,j,0>} = C(L_i) * j, \forall j, 0 \leq j \leq SU_i$ endif
   if $(l_i \in \mathcal{L})$ then $C_{<l_i,0,0>} = +\infty$ else $C_{<l_i,0,0>} == 0$ endif
5. Choose any link l_i, such that row l_i has not been computed, and l_i is incident to a vertex that is also incident to links l_j and l_k such that rows l_j and l_k have been computed. Let v be the node they share.
   $C_{<l_i,p_3,q_3>} = \min(C_{<l_j,p_2,q_2>} + C_{<l_k,p_1,q_1>} + C(l_i) * (p_3 + q_3))$,
      where $valid(p_3,q_3,p_2,q_2,p_1,q_1,v)$ is true.
   If $(l_i \in \mathcal{L})$ then $C_{<l_i,0,0>} = +\infty$ endif
6. If all the links are done, then stop and return $C_{<x,0,0>}$. Else go to step 5.

Figure 5: The algorithm **Tree-Optimal**. This algorithm finds the cost of the optimal solution to the W-MMTCP on a binary tree topology. The function $valid$ is described in the appendix.

6 Experiments and Findings

In this section, we explore the effectiveness of the heuristics presented in the previous two sections. We consider two settings. First, we consider the two existing Internet2 backbone networks, vBNS and Abilene, where we study not only the effectiveness of our heuristics, but also the effect that the routing protocol has on measurement cost. Second, as these networks are rather sparse, we also apply and assess our techniques to a set of denser, randomly generated graphs.

6.1 Internet2 networks

We consider the two Internet2 backbone networks, vBNS [15] and Abilene [16]. Both networks maintain native IP multicast services using the PIM sparse-mode routing algorithm. Since the experimental results on the Abilene network are similar to that on vBNS, in this section, we only present the experimental results from vBNS. The vBNS multicast topology (as of October 25, 1999) is illustrated in [15]. It consists of 160 nodes and 165 edges; each node represents a router and each edge represents a link connecting a pair of routers. The link bandwidths vary between 45M (DS3) and 2.45G (OC48). In our experiments, we assume that the cost of using a link for measurement within one multicast session is inversely proportional to its bandwidth. In addition, we assume that only the leaves in the topology (i.e., node of degree one) can be a sender or a receiver.

6.1.1 Tree heuristic

In section 5 we proposed the algorithm **Tree-Optimal** that is guaranteed to find optimal solutions in polynomial time for any tree topology. We propose and study a heuristic based on that algorithm. This heuristic uses the observation that the topology of networks such as vBNS and Abilene is very close to a tree. Furthermore, the bandwidth of the small number of links that create cycles tends to be high, and thus presumably have low cost.

The heuristic can be applied to any topology, and proceeds in four phases. In the first phase, the topology is converted into a tree by condensing every cycle into a single super-node. In the second phase, the algorithm **Tree-Optimal** is run on the resulting tree. This gives us a set of multicast trees, defined by a list of senders, and for each sender a list of receivers. In the third phase, this set of multicast trees is mapped back to the original topology, so that the same set of senders each send to their respective receivers. This is guaranteed to cover all of the required links that were not condensed into super-nodes, but may leave required links that appear in cycles uncovered. The fourth and final phase uses the fast greedy heuristic to cover any such edge.

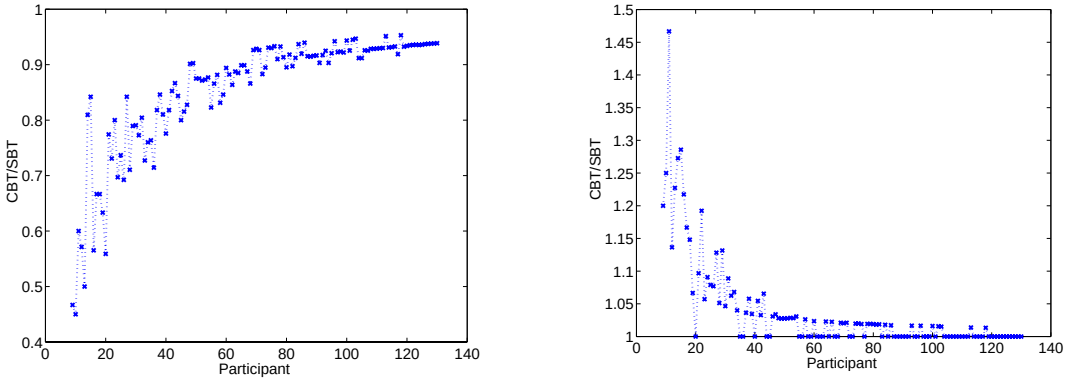
Note that the cost of the solution obtained by **Tree-Optimal** is a lower bound on the cost of the solution to the actual topology. This implies several important properties of this heuristic. Call any link that appears on a cycle in the graph a *cycle link*. If all of the cycle links have zero cost, and no cycle link must be covered, then the tree heuristic is guaranteed to produce an optimal solution. Also, if there are not many cycle links, or they all have relatively small cost, then the solution found by the heuristic will usually be close to the lower bound, and thus close to optimal. The fact that the heuristic produces this lower bound is also useful, as it allows one to estimate how close to optimal the solution produced by the heuristic is.

6.1.2 Core based tree versus source based tree

PIM-SM can be used to either generate a shared tree anchored at a rendezvous point (RP) or to generate source-based trees. In this section, we examine how the choice of tree type impacts the set of links that can be identified, as well as the minimum cost required to cover a set of links. Henceforth, we use the generic terms core and core-based tree in place of RP and shared tree. The core based tree can be either bidirectional where packets can travel both up and down a tree or unidirectional where packets must first be transmitted to the core before they are then transmitted down to the receivers. For simplicity, we only consider bidirectional trees.

Identifiability. We examine how well core based trees can identify links relative to source based trees as a function of the number n, where $n = |R|$. We consider the case where $R = S$. We randomly choose the n nodes from the set of 130 leaves of the vBNS topology with equal probability. Given a set of senders/receivers, we form source based trees and core based trees and count the numbers of edges that each type of tree can cover. We construct two sets of core based trees: one set is constructed under the assumption that all trees can only share a single core, whereas the other set is constructed such that trees can choose any router within a sub-net of the vBNS network.

We here focus on the strong cover and plot the ratio of the number of edges covered by core based trees (CBT) to the number of edges covered by source based trees (SBT) in Figure 6. When there is only one core, source based trees can cover more edges than the core based trees but the difference becomes less as the number of participants increases. When core based trees have multiple cores to choose from, core based trees can cover more edges than source based trees when the number of participants is small. Similar results are observed from experiments where the weak cover is required.

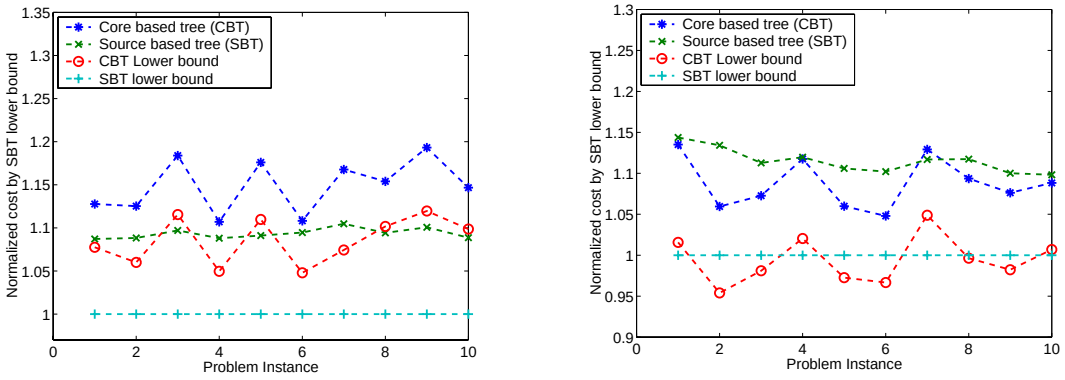


One core Multiple cores

Figure 6: Identifiability: Core Based Trees versus Source Based Trees (Strong Cover)

Cost. Since the tree heuristic provides a lower bound on the cost of a weak cover, we use the tree heuristic to compare the costs of covering a given set of links using core based trees and source based trees. We

assume that every leaf in the vBNS topology can be used as a source and/or a receiver for constructing multicast trees. By varying the links to be covered, we generate ten problem instances. More precisely, we first determine the set of edges which can be covered by both source based trees and core based trees. We then randomly choose 30% of these edges to be covered, where each edge is equally likely to be chosen. For each problem instance, we use the tree heuristic to compute the cost of covering the given set of links if we are required to use source based trees, core based trees with single core and core based trees with multiple cores respectively. Figure 7 plots the costs of these different trees as well as lower bounds on the costs. For each instance, the cost are normalize by the lower bound on the cost of source based trees. We observe that source based trees are cheaper when only a single core is available for constructing core based trees, but if multiple cores are available, then the cost of covering a given set of links is similar for the two routing strategies, with the core based trees being slightly cheaper.



One core Multiple cores

Figure 7: Cost: Core Based Trees versus Source Based Trees (Weak Cover)

6.1.3 Effectiveness of heuristics

In Section 4 we introduced the greedy algorithm and the 0-1 integer programming algorithm for approximating S-MMTCP and W-MMTCP, and described their worst case approximation ratio bounds. In order to approximate the general MMTCP in polynomial time, we also proposed a fast greedy heuristic in Section 4. In this section we study the average performance of these algorithms and heuristics through experiments on the Internet2 backbone networks. Since the topologies of these networks are close to tree topologies, we include the performance of the tree heuristic on Internet2 backbones in our study.

To create a suite of problem instances, we varied the sizes of the source and receiver candidate sets. In addition, for a particular pair of source candidate set and receiver candidate set, we chose the size of the set of links that must be covered to be proportional to the size of the set of links that the source candidate set and the receiver candidate set can identify. For each problem size, we generated 100 random problem instances

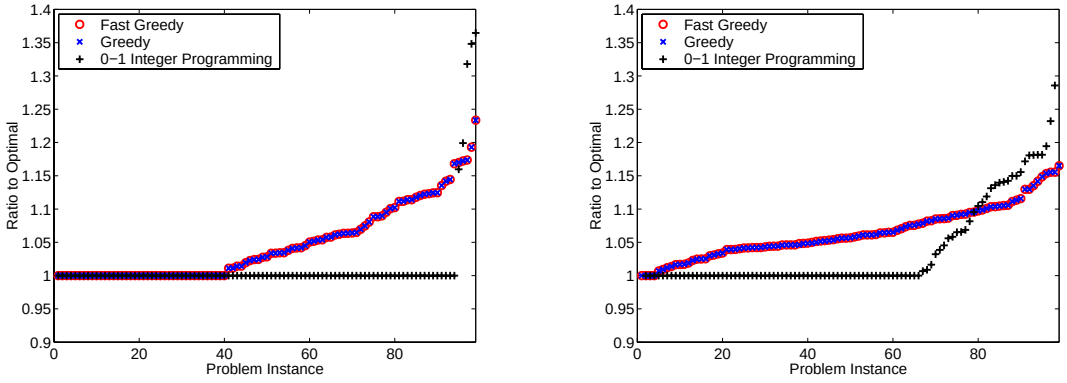
for the vBNS multicast network. For each of these problem instances, we determined the cost of the solution found by each algorithm. We assumed that all the multicast trees have the same fixed initialization cost.

W-MMTCP and S-MMTCP. We ran the algorithms on inputs where the number of source candidates is eight and the number of receiver candidates varies from eight to sixteen. For small problem instances such as these, the optimal solutions can be computed for these problem sizes using exhaustive search, and this can be used to check the quality of the approximation results. For both W-MMTCP and S-MMTCP, we used the 0-1 integer programming, greedy and fast greedy algorithms to approximate the 100 problem instances on vBNS for each of the problem size. In addition, we used the tree heuristic to approximate the W-MMTCP. We compare the performance of the algorithms for S-MMTCP in Figure 8 and W-MMTCP in Figure 9. In both figures, the ratio of the solutions found by the approximation algorithm to the optimal solutions is plotted. For each approximation algorithm, we sort the ratios in ascending order. Thus, for example, problem instance 1 for each algorithm represents the instance where that algorithm performed the closest to optimal, and may correspond to different inputs for the different algorithms. We present plots for two different problem sizes: 8 sources and 8 receivers, as well as 8 sources and 16 receivers.

In Figure 8, it is surprising to see that the fast greedy algorithm produces the same solution as the greedy algorithm and that the 0-1 integer programming algorithm yields the optimal solution on most inputs when the problem size is small. As the problem size increases, the 0-1 integer programming is less likely to produce the optimal solution and the difference between the fast greedy and the optimal seems to increase slowly.

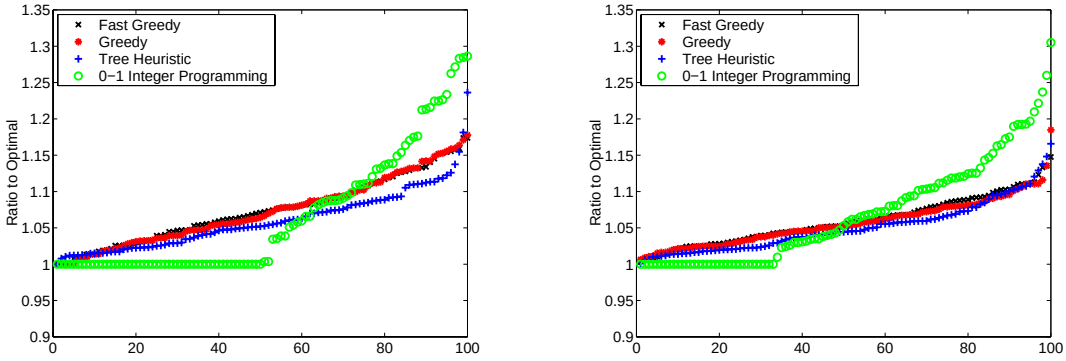
In the case of approximating W-MMTCP, the tree heuristic out-performs both the greedy algorithm and the fast greedy algorithm on most problem instances. The quality of the 0-1 integer programming algorithm decreases as the problem size increases. The results from the fast greedy are only slightly worse than those from the greedy algorithm. The difference between the fast greedy algorithm and greedy algorithm seems to change very slowly as the problem size increases.

M-MMTCP The complexity of the exhaustive search algorithm to compute the optimal solution for the M-MMTCP is high. Thus, we can only compute the optimal solution for very small problem sizes. Instead of showing the ratio of the solution returned by our heuristics to the optimal solution, we focus on comparing the results from the greedy algorithm and the fast greedy heuristic. In Figure 10, we plot the ratio of solutions from the fast greedy heuristic and the greedy algorithm. We sort the ratios in ascending order. The difference in the quality of solution between the greedy algorithm and the fast greedy heuristic seems to increase slowly as the problem size increases. The fast greedy heuristic can produce better results than the greedy algorithm on a small number of instances.



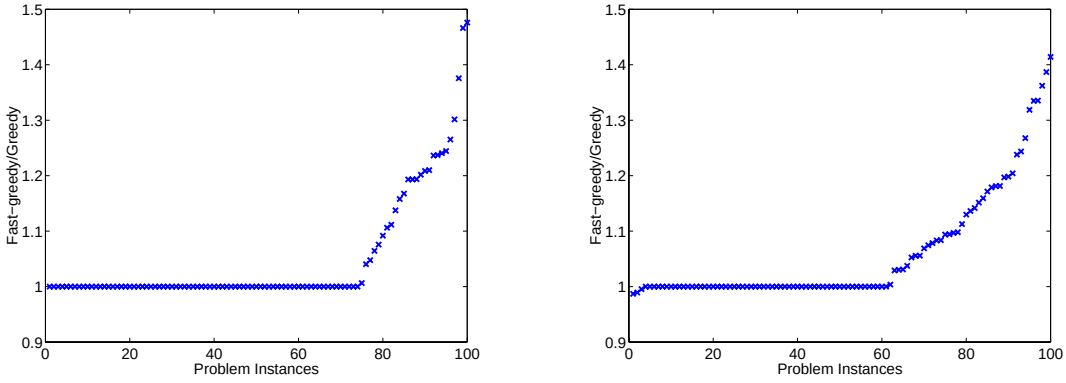
8 sources and 8 receivers 8 sources and 16 receivers

Figure 8: Comparison of approximation algorithms for the S-MMTCP on vBNS.



8 sources and 8 receivers 8 sources and 16 receivers

Figure 9: Comparison of approximation algorithms for the W-MMTCP on vBNS.



8 sources and 8 receivers 8 sources and 16 receivers

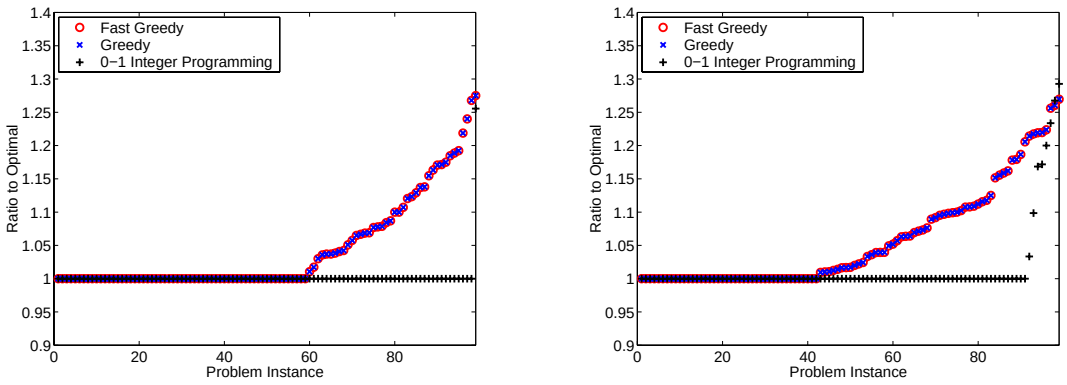
Figure 10: Comparison of approximation algorithms for the M-MMTCP on vBNS.

6.2 Experiments on dense networks

Both vBNS and Abilene are quite sparse, i.e., each network only contains a very small number of additional edges than a tree topology containing the same number of nodes. In this section, we investigate how our

algorithms perform on denser network than vBNS and Abilene. Unfortunately, we had no such multicast topologies available to us. Instead, we make use of randomly generated topologies.

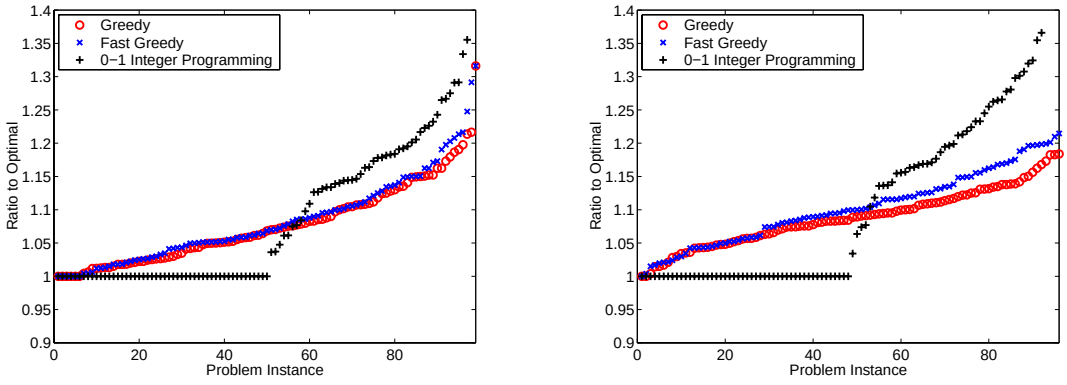
We generated ten 100-node transit-stub undirected graphs using GT-ITM (GT internetwork topology model). For more details about the transit-stub network model, please refer to [14]. The average out-degree is in the range of $[2.5, 5]$. We assigned two costs to each edge in the graphs, one for each direction. These cost are uniformly distributed in $[2, 20]$. By randomly picking the source candidate set, receiver candidate set and to-be covered set of links and then assigning costs to the edges, we generated ten problem instances for each graph. We ran all algorithms on a total of 100 problem instances and compared their performance. We assumed that all the multicast trees have the same fixed initialization cost.



8 sources and 8 receivers

8 sources and 16 receivers

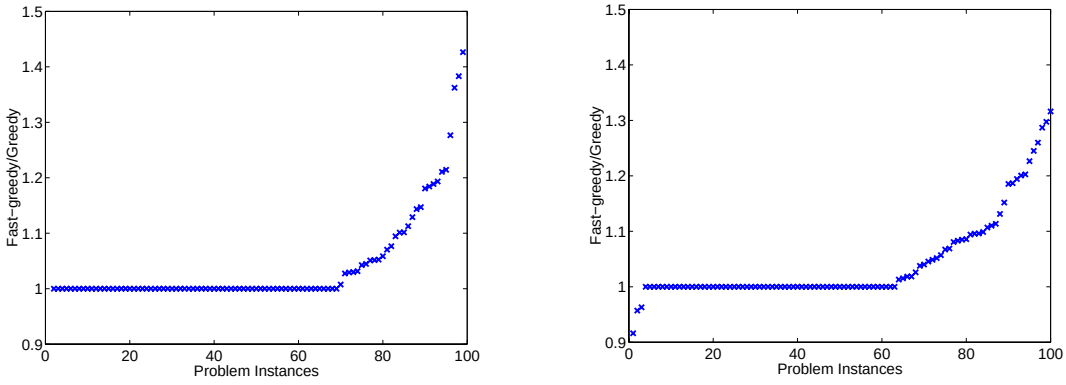
Figure 11: Comparison of approximation algorithms for the S-MMTCP on 100-node transit-stub.



8 sources and 8 receivers

8 sources and 16 receivers

Figure 12: Comparison of approximation algorithms for the W-MMTCP on vBNS on 100-node transit-stub.



8 sources and 8 receivers

8 sources and 16 receivers

Figure 13: Comparison of approximation algorithms for the M-MMTCP on vBNS on 100-node transit-stub.

6.2.1 S-MMTCP and W-MMTCP

We ran the algorithms on inputs where the number of source candidates is eight and the number of receiver candidates varies from eight to sixteen. We used the 0-1 integer programming, greedy and fast greedy algorithms to approximate the 100 problem instances for each of the problem sizes. We compare the performance of the algorithms for the S-MMTCP in Figure 11 and the W-MMTCP in Figure 12. In both figures, the ratio of the solution found by the approximation algorithms to the optimal solution is plotted. For each approximation algorithm, we sorted the ratios ascendantly.

In Figure 11, it is surprising to see that the fast greedy algorithm produces the same solution as the greedy algorithm and the 0-1 integer programming yields the optimal solution on most inputs. As the problem size increases, the greedy and fast greedy algorithm are less likely to produce the optimal solution.

In the case of approximating W-MMTCP, the 0-1 integer programming algorithm yields the optimal solution on about half the problem instances. However, it yields worse results than the greedy and fast greedy algorithm for about forty percent of the problem instances. The results from the fast greedy are slightly worse than those from the greedy algorithm in most of cases. The difference between the fast greedy algorithm and greedy algorithm seems to increase very slowly as the problem size increases.

6.2.2 M-MMTCP

We also ran the algorithms for the M-MMTCP on inputs where the number of source candidates is eight and the number of receiver candidates varies from eight to sixteen. We compare the results from the greedy algorithm and the fast greedy heuristic. In Figure 13, we plot the ratio of solutions from the fast greedy heuristic and the greedy algorithm. We sort the ratios in ascending order. The fast greedy heuristic yields the same results as the greedy algorithm on more than sixty percent of the problem instances. In approximately five

percent of the problem instances, the fast greedy heuristic yields better solutions than the greedy algorithm. The difference in the quality of solution between the greedy algorithm and the fast greedy heuristic seems to increase slowly as the problem size increases.

7 Conclusions

In this paper we focussed on the problem of selecting trees from a candidate set in order to cover a set of links of interest. We identified three variation of this problem according to the definition of cover and addressed two questions for each of them:

- is it possible to cover the links of interest using trees from the candidate set?
- if the answer to the first question is yes, what is the minimum set of trees that can cover the links?

We proposed computationally efficient algorithms for the first of these questions. We also established, with some exceptions, that determining the minimum cost set of trees is a hard problem. Moreover, it is a hard problem even to develop approximate solutions. One exception is when the underlying topology is a tree in which case we present efficient dynamic programming algorithms for two of the covers. We also proposed several heuristics and showed through simulation that a greedy heuristic that combines trees with three or fewer receivers performs reasonably well.

Last, we applied our methods to the vBNS and Abilene networks and observed

- a heuristic based on the tree algorithm gives excellent results,
- the cost of the set of trees required to cover a set of links is relatively insensitive to the use of either a core based tree or a source based tree algorithm.

References

[1] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. Lo Presti, S.B. Moon, V. Paxson. “The use of end-to-end multicast measurements for characterizing internal network behavior,” *IEEE Communications Magazine*, **38**(5)152-159, May 2000.

[2] K. Almeroth, L. Wei, D. Farinacci. “Multicast Reachability Monitor (MRM)” IETF Internet Draft, Oct. 1999.

[3] K. Almeroth. “The evolution of multicast,” *IEEE Network*, **14**(1)10-21, Jan. 2000.

[4] V. Chvátal “A greedy heuristic for the set-covering problem” *Mathematics of Operations Research*, 4(3)233-235, Aug. 1979

[5] M. Coates, R. Nowak. “Network loss inference using unicast end-to-end measurement”, *Proc. ITC Conf. IP Traffic, Modeling and Management*, Sept. 2000.

[6] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C-G Liu, L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Transactions on Networking*, 4, 2, 153–162, April 1996.

[7] C. Diot, B.N. Levine, B. Lyles, H. Kassem, D. Balensiefen. "Deployment issues for the IP multicast service and architecture," *IEEE Network*, **14**(1)78-89, Jan. 2000.

[8] N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley. "Inferring Link Loss Using Striped Unicast Probes", to appear in *Proc. INFOCOM01*.

[9] T. Pusateri, "Distance Vector Multicast Routing Protocol", Internet Draft, draft-ietf-idmr-dvmrp-v3-04.ps, Feb. 1997.

[10] A. Reddy, R. Govindan, D. Estrin. "Fault isolation in multicast trees," *Proc. SIGCOMM 2000*, Sept. 2000.

[11] Y. Shavitt, X. Sun, A. Wool, B. Yener. "Computing the Unmeasured: An Algebraic Approach to Internet Mapping" to appear in *Proc. INFOCOM01*.

[12] A. Srinivasan "Improved approximation of packing and covering problem " *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing* 268-276, Las Vegas, Nevada, June 1995

[13] J. Walz, B.N. Levine. "A practical multicast monitoring scheme," U. Massachusetts Computer Science Technical Report 30-2000, June 2000.

[14] Ellen W. Zegura, Ken Calvert and S. Bhattacharjee " How to model an internetwork" *Proceedings of IEEE Infocom '96, San Francisco, CA.*

[15] http://www.vbns.net/netmaps/multicast.html

[16] http://www.abilene.iu.edu/images/ab-mcast.pdf

A Details of the algorithm Tree-Optimal

We first provide the proof of Theorem 5, restated here for convenience.

Theorem 5 *The algorithm* **Tree-Optimal** *finds the optimal cost of a solution to the W-MMTCP in any binary tree in $O(|E||S|^6)$ steps.*

Proof: Each entry in the table in row l can be computed by examining $|S|^4$ pairs of entries in two other rows m and n, and performing a constant number of steps per pair of entries examined. The bound on the running time follows from the fact that there are $|E|(|S|+1)^2$ table entries. In order to prove that the algorithm returns the correct answer, we show that for all l, i, j, including $l = x, i = j = 0$, the correct value of $C_{<l,i,j>}$ is computed. We show this is true for the row corresponding to any link l by a reverse induction on the distance that l is from the link x. The base cases are the values computed for the links attached to leaves of the tree N, which can be seen trivially to be correct.

For the inductive step, assume that we want to compute the row associated with the link l, as depicted in Figure 4. By the inductive hypothesis, we can assume that the rows corresponding to the links m and n are correct. When determining the correct value for entry $C_{<m,i_m,j_m>}$, it was assumed that flows going over the link m originated or ended at the node v. However, the cost of these flows going over links in the subtree ST_m is not affected by whether or not these flows also pass over link l and/or links in the subtree ST_n. Therefore, for any values of i and j, the optimal cost of all flows going over links in the subtree ST_m in the best solution for the tree ST_l that sends j_m flows from v across m and receives i_m flows to v across m is

$C_{<m,i_m,j_m>}$. Similarly, the optimal cost of all flows going over links in the subtree ST_n in the best solution for the tree ST_l that sends j_n flows from v across n and receives i_n flows to v across n is $C_{<n,i_n,j_n>}$. Since the algorithm **Tree-Optimal** tries all possible values of flows across links m and n, the entry $C_{<l,i,j>}$ is correct, which completes the induction. ■

We also provide some additional details of the algorithm **Tree-Optimal**. In particular, we describe how to determine, for a set of three links all incident to the same node, whether or not a given specification of flows along each of the links leads to a valid multicast flow. We assume that we are given 6 values $l_{in}, l_{out}, m_{in}, m_{out}, n_{in}$, and n_{out} that describe the number of flows in to and out of a node v along the three links l, m, and n, all incident to node v. We also assume that we are given a description of the node v that allows us to determine whether or not v is a sender and/or a receiver.

We here provide an algorithm that requires determining if an integer programming problem has a feasible solution. Standard techniques can solve an integer programming problem in time that is exponential in the number of constraints. The number of constraints in the algorithm is constant, and thus we can determine if the values of the flows lead to a valid set of multicast flows through the node v in constant time. We point out that if this integer program were used as a subroutine for the algorithm **Tree-Optimal**, this would lead to large constants hidden in the asymptotic notation. However, a more complicated technique can remove these large constants.

Function: $valid(l_{in}, l_{out}, m_{in}, m_{out}, n_{in}, n_{out}, v)$

1. Set s_v to 1 if v is source, otherwise set s_v to 0.
2. Let N_{xy} denote a possible number of flows which first traverse link x and then link y. Define the following constrains:

Set I

$$N_{lm} + N_{ln} \geq l_{in}$$
$$N_{ml} + N_{mn} \geq m_{in}$$
$$N_{nl} + N_{nm} \geq n_{in}$$

Set II

$$N_{lm} + N_{nm} \leq m_{out} \leq N_{lm} + N_{nm} + s_v$$
$$N_{ln} + N_{mn} \leq n_{out} \leq N_{ln} + N_{mn} + s_v$$
$$N_{nl} + N_{ml} \leq l_{out} \leq N_{nl} + N_{ml} + s_v$$
$$0 \leq N_{lm} \leq l_{in}$$
$$0 \leq N_{ln} \leq l_{in}$$
$$0 \leq N_{mn} \leq m_{in}$$
$$0 \leq N_{ml} \leq m_{in}$$
$$0 \leq N_{nl} \leq n_{in}$$
$$0 \leq N_{nm} \leq n_{in}$$

3. If v is not a receiver, return true if there is a feasible integer solution for constraint sets I and II.
4. If v is a receiver, return true if there is a feasible integer solution for constraint set II.

Figure 14: How to check if the number of flows into and out of a node is valid

Inference of Internal Loss Rates in the MBone*

Ramón Cáceres, N.G. Duffield
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932, USA
{ramon,duffield}@research.att.com

Sue B. Moon, Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
{sbmoon,towsley}@cs.umass.edu

June 25, 1999

Abstract

We present MBone experiments that validate an end-to-end measurement technique we call MINC, for Multicast Inference of Network Characteristics. MINC exploits the performance correlation experienced by multicast receivers to infer loss rates and other attributes of internal links in a multicast tree. MINC has two important advantages in the Internet context: it does not rely on network collaboration and it scales to very large measurements. In previous work, we laid the foundation for MINC using rigorous statistical analysis and packet-level simulation. Here, we further validate MINC by comparing the loss rates on internal MBone tunnels as inferred using our technique and as measured using the `mtrace` tool. Inferred values closely matched directly measured values – differences were usually well below 1%, never above 3%, while loss rates varied between 0 and 35%.

1 Introduction

As the Internet grows in size and diversity, its internal performance becomes harder to measure. Any one organization has administrative access to only a small fraction of the network's internal nodes, while commercial factors often prevent organizations from sharing internal performance data. End-to-end measurements using unicast traffic do not rely on administrative privileges, but it is difficult to infer link-level performance from them and they require large amounts of traffic to cover multiple paths. There is a need for practical and efficient procedures that can take an internal snapshot of a significant portion of the network.

We have developed a measurement technique that addresses these problems. *Multicast Inference of Network Characteristics* (MINC) [11] uses end-to-end multicast traffic as measurement probes. It exploits the inherent correlation in performance observed by multicast receivers to infer the loss rate and other attributes of paths between branch points in a multicast routing tree. These measurements do not rely on administrative access to internal nodes since they are done between end hosts. In addition, they scale to large networks because of the bandwidth efficiency of multicast traffic.

The intuition behind packet loss inference is that the event that a packet has reached a given internal node in the tree can be inferred from the packet's arrival at one or more receivers descended from that node. Conditioning on this event, we can determine the probability of successful transmission to and beyond the given node. Consider, for example, a simple multicast tree with a root node (the source), two leaf nodes (the left and right receivers), a link from the source to a branch point (the shared link), and a link from the branch point to each of the receivers (the left and right links). The source sends a stream of sequenced multicast packet through the tree to the two

*This work was sponsored in part by DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.

receivers. If a packet reaches either receiver, we can infer that the packet reached the branch point. Thus the ratio of the number of packets that reach both receivers to the number that reached only the right receiver gives an estimate of the probability of successful transmission on the left link. The probability of successful transmission on the other links can be found by similar reasoning.

It is not immediately clear whether this technique applies to more than just binary trees or whether it enjoys desirable statistical properties. In previous work [2], we extended this technique to general trees and showed that the estimate is consistent, that is, it converges to the true loss rates as the number of probes grows. More specifically, we developed a *Maximum Likelihood Estimator* (MLE) for internal loss rates in a general tree assuming independent losses across links and across probes. We derived the MLE's rate of convergence and established its robustness with respect to certain violations of the independence assumption. We also validated these analytical results using the `ns` simulator [15]. We give a brief account of these results in Section 2.2.

In more recent work [3], we explored the accuracy of our packet loss estimation under a variety of network conditions. Again using `ns` simulations, we evaluated the error between inferred and actual loss rates as we varied the network topology, propagation delay, packet drop policy, background traffic mix, and probe traffic type. We found that, in all cases, MINC accurately inferred the per-link loss rates of multicast probe traffic.

In this paper, we further validate MINC through experiments under real network conditions. We used a collection of end hosts connected to the MBone, the multicast-capable subset of the Internet [10]. We chose one host as the source of multicast probes and used the rest as receivers. We then made two types of measurements simultaneously: end-to-end loss measurements between the source and each receiver, and direct loss measurements at every internal node of the multicast tree. Finally, we ran our inference algorithm on the results of the end-to-end measurements, and compared the inferred loss rates to the directly measured loss rates. Across all our experiments, the inferred values closely matched the directly measured values. The differences between the two were usually well below 1%, never above 3%, while loss rates varied between 0 and 35%. Furthermore, the inference algorithm converged well within 2-minute, 1200-probe measurement intervals.

The rest of this paper is organized as follows: Section 2 describes our experimental methodology; Section 3 presents our experimental results; Section 4 discusses our ongoing work; Section 5 surveys related work; and Section 6 offers some conclusions.

2 Experimental Methodology

During each of our MBone experiments, we had a source send a stream of sequenced packets to a collection of receivers while we made two types of measurement at each receiver. At the source, we used our `mgen` traffic generation tool to send one 40-byte packet every 100 milliseconds to a specific multicast group. The resulting traffic stream placed less than 4 Kbps of load on any one MBone link. We reserved multicast address 224.2.130.64 and port 22778 for our experiments using the `sdr` session directory tool [21]. At each receiver, we ran the `mtrace` [13] and `mbat` [9] tools to gather statistics about traffic on this multicast group. Below we describe our use of `mtrace` and `mbat` in more detail.

2.1 Direct measurements

`mtrace` traces the *reverse* path from a multicast source to a receiver. It runs at the receiver and issues trace queries that travel hop-by-hop up the multicast tree towards the source. Each router along the path responds to these queries with information about traffic on the specified multicast group as seen by that router, including counts of incoming and outgoing packets. `mtrace` calculates packet losses on a link by comparing the packet counts returned by the two routers at either end of the link.

In each of our experiments, we collected `mtrace` statistics for consecutive two-minute intervals over the course of one hour. We ran a separate instance of `mtrace` for each interval. Each `mtrace` run issued a trace query at the beginning of the interval and another query at the end. We thus measured link-level loss rates for all thirty intervals in one hour as shown in Figures 2 – 4. These intervals are not exactly two

Physical location	Abbreviation
AT&T Labs – Research, Florham Park, New Jersey	AT&T
Carnegie Mellon University, Pittsburgh, Pennsylvania	CMU
Georgia Institute of Technology, Atlanta, Georgia	GaTech
University of California, Berkeley, California	UCB
University of Kentucky, Lexington, Kentucky	UKy
University of Massachusetts, Amherst, Massachusetts	UMass
University of Southern California, Los Angeles, California	USC
University of Washington, Seattle, Washington	UWash

Table 1: End hosts used during our MBone experiments.

Physical location	Abbreviation
Atlanta, Georgia	GA
Cambridge, Massachusetts	MA
San Francisco, California	CA
West Orange, New Jersey	NJ

Table 2: Routers at multicast branch points during our representative MBone experiment.

minutes long due to delays incurred in collecting responses to the queries. We recorded timestamps for the actual beginning and end of each `mtrace` run to help synchronize our inference calculations to these direct measurements.

We chose to measure two-minute intervals based on our previous experience with MINC. Our simulations have shown that the statistical inference algorithm at the heart of MINC converges to true loss rates after roughly 1,000 observations [2]. Given the 100 milliseconds between probes in our MBone experiments, two minutes allow for 1,200 probes between measurements. As shown in Figure 5, 1,200 probes were indeed enough for MINC to converge.

It is important to note that `mtrace` does not scale to measurements of large multicast groups if used in parallel from all receivers as we describe here. Parallel `mtrace` queries come together as they travel up the tree. Enough such queries will overload routers and links with measurement traffic. We used `mtrace` in this way only to validate MINC on relatively small multicast groups before we move on to use MINC alone on larger groups.

2.2 Statistical inference

MINC works on *logical* multicast trees. A logical tree is one where all nodes, except the root and the leaves, have at least two children. A physical tree can be converted into a logical tree by deleting all nodes, other than the root, that have only one child and then collapsing the links accordingly. A link in a logical tree may thus represent multiple physical links. This conversion is necessary because inference based on correlation among receivers cannot distinguish between two physical links unless these links lead to two different receivers. Henceforth when we speak of trees we will be speaking of logical trees.

2.2.1 Inference algorithm

Our model for loss on a multicast tree assumes that packet loss is independent across different links of the tree, and independent between different probes. With these assumptions, the loss model is specified by associating a probability α_k with each node k in the tree. α_k is the probability that a packet is transmitted successfully across the link terminating at node k, given that it reaches the parent node $p(k)$ of k.

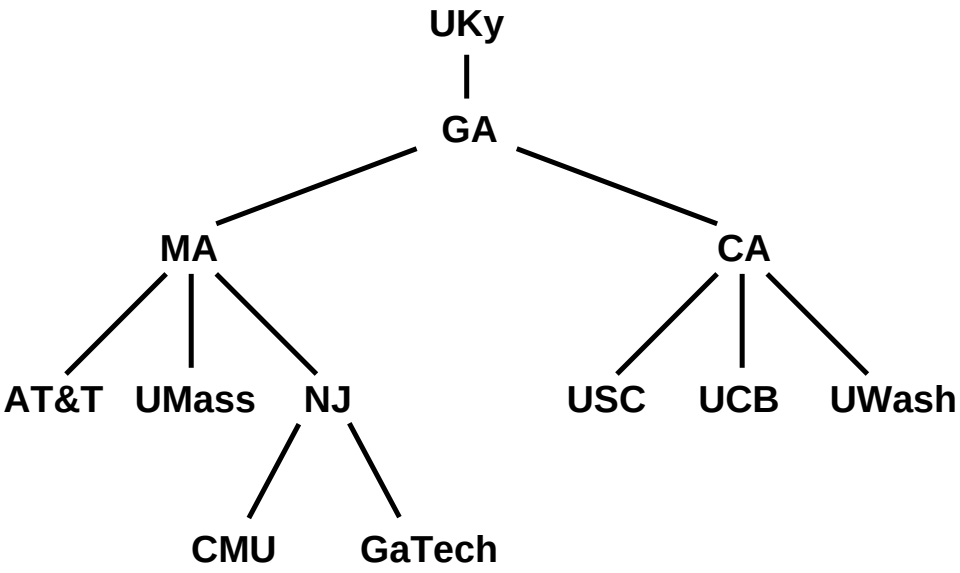


Figure 1: Multicast routing tree during our representative MBone experiment.

When a probe is transmitted from the source, we can record the outcome as the set of receivers the probe reached. The loss inference algorithm is based on probabilistic analysis that allows us to express the α_k directly in terms of the *expected* frequencies of such outcomes. More precisely, for each node k let γ_k denote the probability of the outcome that a given packet reaches at least one receiver that has k as an ancestor in the tree. Let A_k denote the probability that a given packet reaches the node k, i.e., $A_k = \alpha_k \alpha_{k_1} \alpha_{k_2} \ldots \alpha_{k_m}$ where $k_1, k_2, \ldots, k_m$ is the chain of m adjacent nodes leading back from node k to the root of the tree. Then it can be shown that A_k satisfies

$$(1 - \gamma_k / A_k) = \prod_{j \in c(k)} (1 - \gamma_j / A_k) \qquad (1)$$

where the product is taken over all nodes j in $c(k)$, the set of children of the node k. It was shown in [2] that under generic conditions the A_k can be recovered uniquely through (1) if the γ are known. The α_k can in turn be recovered since $\alpha_k = A_k / A_{p(k)}$. Generally, finding A_k requires numerical root-finding for (1). In the special case of a node k with two offspring j and j', (1) can be solved explicitly:

$$A_k = \frac{\gamma_j \gamma_{j'}}{\gamma_j + \gamma_{j'} - \gamma_k} \qquad (2)$$

Suppose that in place of the γ_k in (1), we use the *actual* frequencies $\widehat{\gamma}_k$ with which n probes reach at least one receiver with ancestor k. We denote the corresponding solutions to (1) by $\widehat{A}_k$ and estimate the link probabilities by $\widehat{\alpha}_k = \widehat{A}_k / \widehat{A}_{p(k)}$. The calculation of the $\widehat{\gamma}_k$ is achieved though a simple recursion as follows. Define new variables $Y_k(i)$ as function of the measured outcomes of n probes by

$$Y_k(i) = \begin{cases} 1 & \text{if probe } i \text{ reaches node } k \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

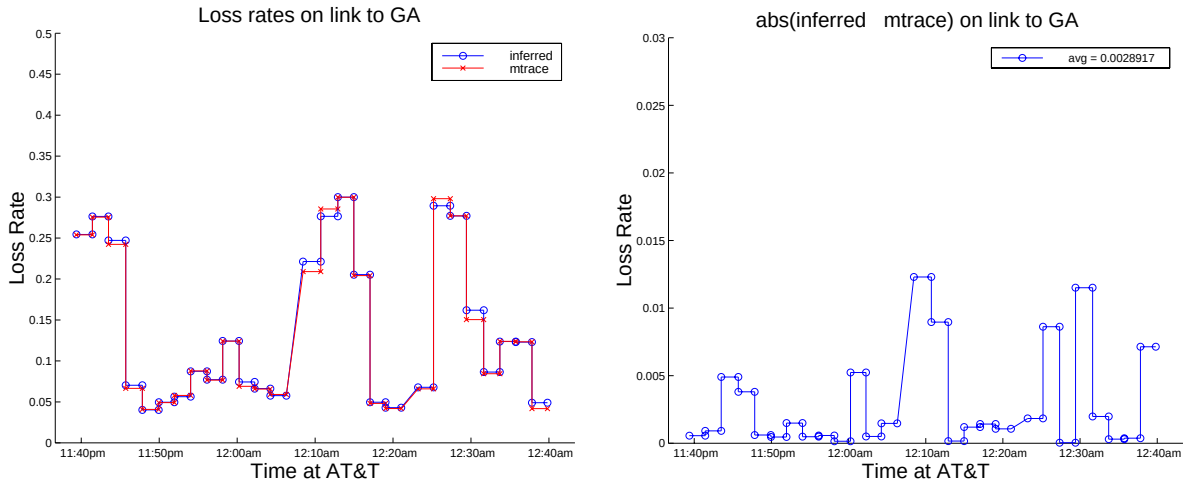
if k is a leaf node, and

$$Y_k(i) = \max_{j \in c(k)} Y_j(i) \qquad (4)$$

otherwise. Then

$$\widehat{\gamma}_k = \frac{1}{n} \sum_{i=1}^{n} Y_k(i) \qquad (5)$$

We showed in [2] that the estimator $\widehat{\alpha}_k$ enjoys two useful properties: (i) *consistency*: $\widehat{\alpha}_k$ converges to the true value α_k almost surely as the number of probes n grows to infinity, and (ii) *asymptotic normality*: the distribution of the normalized difference $\sqrt{n}(\widehat{\alpha}_k - \alpha_k)$ converges to a normal distribution as n grows to infinity. We also investigated in [2] the effects of correlations that violate the independent



(a) Inferred vs. directly measured loss rates

(b) Difference between the two

Figure 2: Loss rates on link to GA when running `mtrace` from AT&T. The two sets of loss rates agreed closely over a wide range of values. Differences remained below 1.5% while loss rates varied between 4 and 30%.

loss assumptions. Consistency is preserved under a large class of temporal correlations, although convergence of the estimates with n can be slower. Spatial correlations perturb the estimate continuously, in that small correlations lead to small inconsistencies. When losses on sibling links are correlated the perturbation is a second-order effect, in that the degree of inconsistency depends not on the size of the correlations, but on the degree to which they change across the tree.

Our earlier papers on MINC [2, 3] contain a detailed description and analysis of the above inference algorithm, including rules to handle special cases of the data in which the generic conditions required for the existence of solutions to (1) fail. In the interests of brevity, we omit these details from this paper.

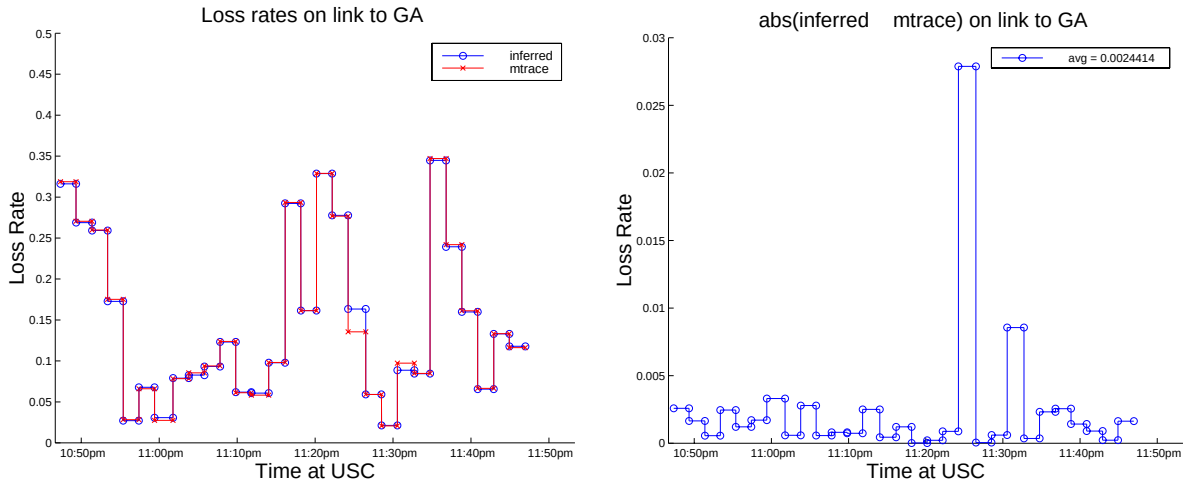
2.2.2 Inference calculations

We encoded our loss inference algorithm in a program called `infer`. `infer` takes two inputs: a description of the tree topology and a description of the end-to-end losses experienced by each receiver. It produces as output the estimated loss rates on every link in the tree.

We determined the tree topology by combining the `mtrace` output from all the receivers. Along with packet counts, `mtrace` reports the domain name and IP address of each router on the path from the source to a receiver. We built a complete multicast tree by looking for common routers and branch points on the paths to all the receivers. The topology of the MBone is relatively static due to that network's current reliance on manually configured IP-over-IP tunnels. These tunnels are themselves logical links that may each contain multiple physical links. We verified that the topology remained constant during our experiments by inspecting the path information we obtained every two minutes from `mtrace`.

We measured end-to-end losses using the `mbat` tool. `mbat` runs at a receiver, subscribes to a specified multicast group, and collects a trace of the incoming packet stream, including the sequence number and arrival time of each packet. We ran `mbat` at each receiver for the duration of each experiment. At the conclusion of an experiment, we transferred the `mbat` traces and `mtrace` output from all the receivers to a single location.

There we ran the loss inference algorithm on the same two-minute intervals on which we collected `mtrace` measurements. For each receiver, we used the timestamps for the beginning and end of `mtrace` measurements to segment the `mbat` traces into corresponding two-minute subtraces. Then we ran `infer` on each two-minute interval and compared



(a) Inferred vs. directly measured loss rates

(b) Difference between the two

Figure 3: Loss rates on link to GA when running `mtrace` from USC. These measurements span different two-minute intervals than those from AT&T (see Fig. 2) because of clock asynchrony. Nevertheless, inferred and directly measured loss rates agreed closely. Differences were usually below 0.5%, never above 3%, while loss rates varied between 2 and 35%.

the inferred loss rates with the directly measured loss rates. We discuss the results in the next section.

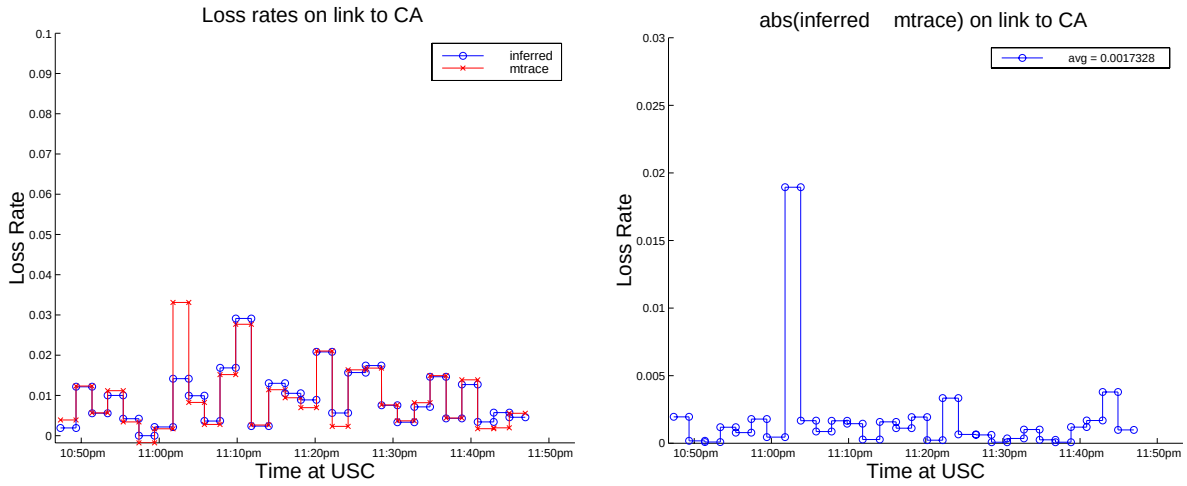
3 Experimental Results

We performed a number of MBone experiments using different multicast sources and receivers, and thus different multicast trees. Inferred loss rates agreed closely with directly measured loss rates throughout our experiments. Here we discuss results from a representative experiment on August 26, 1998. Tables 1 and 2 list the end hosts and branch routers involved in this experiment, while Figure 1 shows the resulting multicast tree.

Figure 2 shows that inferred and directly measured loss rates agreed closely despite a link experiencing a wide range of loss rates. In this case, loss rates as measured by `mtrace` varied between 4 and 30%. Nevertheless, differences between inferred and directly measured loss rates remained below 1.5%.

Figures 2 – 4 all show that inferred and directly measured loss rates agreed closely despite imperfect synchronization between `infer` and `mtrace` intervals. The two sets of intervals do not always match because of variable network delays. The timestamps for the beginning and end of `mtrace` intervals are recorded before a trace query is issued and after a trace query returns, both according to the clock at the relevant receiver. However, the corresponding packet counts are recorded at the time the trace query arrives at each router. Therefore, although the `infer` intervals are derived from the `mtrace` intervals using the same receiver clock, the inference is not always applied to exactly the same 1,200 probe packets as the direct loss measurement. Nevertheless, differences between inferred and directly measured loss rates across Figures 2 – 4 were usually well below 1%, never above 3%.

Along the same lines, Figures 2 and 3 together show that inferred and directly measured loss rates agreed closely for different two-minute intervals on the same link. We have multiple sets of `mtrace` measurements for links shared by multiple receivers, one set for each receiver. In these cases, we can run `infer` on different sets of intervals corresponding to the different sets of `mtrace` intervals. `mtrace` intervals are different for each receiver because of clock asynchrony between receivers and because of the variable network delays discussed above. Nevertheless, differences between inferred and directly measured loss rates across Figures 2 and 3 remained below 3%.



(a) Inferred vs. directly measured loss rates

(b) Difference between the two

Figure 4: Loss rates on link to CA when running `mtrace` from USC. CA experienced an order of magnitude lower loss rates than GA (see Figs. 2 and 3). Nevertheless, inferred and directly measured loss rates agreed closely. Differences were usually below 0.5%, never above 2%, while loss rates varied between 0 and 4%.

Figure 4 shows that inferred and directly measured loss rates agreed closely even for links with very low loss rates. In this case, loss rates varied between 0 and 4%, an order of magnitude lower than the loss rates in Figure 2. Nevertheless, differences between inferred and directly measured loss rates were usually below 0.5%, never above 2%.

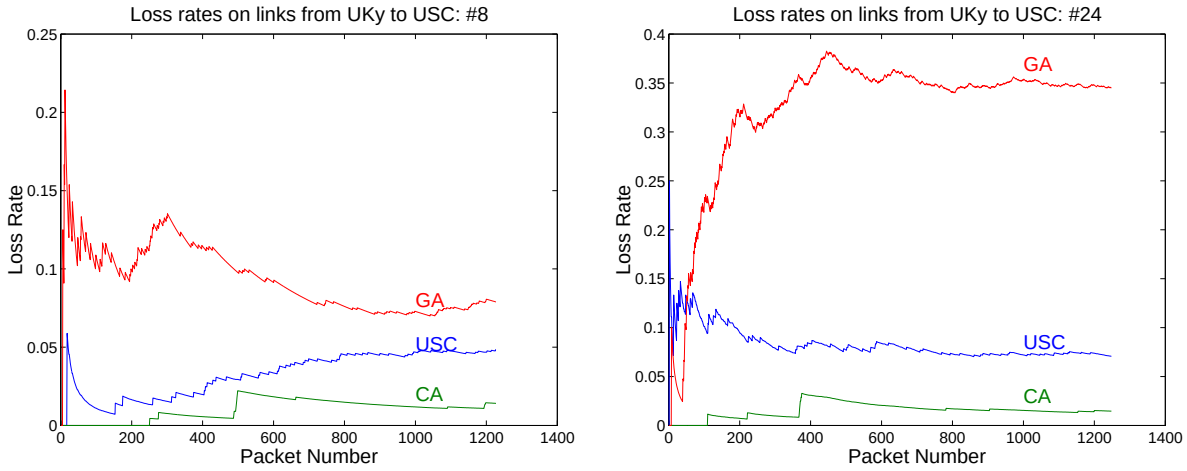
Finally, Figure 5 shows that the inference algorithm converged quickly to the desired loss rates. Each inferred loss rate reported in Figures 2 – 4 is the value calculated by `infer` at the end of the corresponding 2-minute, 1200-probe measurement interval. However, `infer` outputs a loss rate value for every probe. Figure 5 reports these intermediate values. As shown, inferred loss rates stabilized well before a measurement intervals ends. Our algorithm converged after fewer than 800 probes for all links and all measurement intervals in our experiments.

4 Ongoing Work

The results reported in the previous section suggest that it is possible to characterize link-level loss based on end-to-end multicast measurements. However, a number of issues need to be resolved before the technology can be deployed and made available for general use.

First, there is the question of how end-to-end multicast measurements are to be generated and collected. We are pursuing two approaches for addressing this question. The first is to add a multicast probe capability to an existing measurement infrastructure. We are working with the National Internet Measurement Infrastructure (NIMI) [14] project to do exactly this. Currently NIMI permits users to schedule a variety of unicast end-to-end measurements between NIMI platforms and to download the traces to a site of their choosing. We are augmenting this capability to permit the scheduled execution of multicast end-to-end measurements followed by a distribution of the traces. Once we have accomplished this, we will also be able to design and execute a more extensive set of experiments to validate the inference techniques against `mtrace`.

There is one disadvantage with the above approach, namely that the set of links that can be covered is limited by the number and placement of NIMI nodes. We are investigating a second approach that has the potential of addressing that problem. The basic idea is to gather end-to-end loss information for multicast applications, such as teleconferencing and continuous media streaming, that already exist in the network. This is possible when the application uses the Real-Time Transport Protocol (RTP) and its as-



(a) Eighth measurement interval

(b) Twenty-fourth measurement interval

Figure 5: Inferred loss rates on the three links between UKy and USC (i.e., the links to GA, CA, and USC) during individual 2-minute, 1200-probe measurement intervals. The inference algorithm converged well before the measurement interval ended for all links during all measurement intervals.

sociated control protocol, RTCP [20]. Currently, applications using these protocols require receivers to multicast loss and delay information to each other. Currently the loss information is limited, consisting of short-term and long-term loss rates, and is not adequate for our inference techniques.

We plan to evaluate different ways of augmenting these RTCP loss reports to provide sufficient information to infer link-level loss behavior. This would allow a measurement node anywhere in the network to monitor and record the RTCP loss reports for various applications. This measurement node could identify the topology of an application using the third-party measurement feature of `mtrace` [13], then apply the inference methodology to obtain the link-level behavior. The hope is that the number of participants in these applications will be sufficiently large to allow a measurement node to estimate the loss behavior of a large portion of the links in the network.

A second important deployment issue concerns the need to know the topology of the multicast tree in order to apply our techniques. Our current experiments use the multicast tree topology discovered from executing `mtrace`. Recent work has shown that algorithms based on link-level loss estimators for binary trees can be used to infer the topology of multicast trees. Topology inference of binary and general trees was proposed in [19] and [4], respectively. Although not reported here, the latter algorithms are able to infer the trees described in Section 3 with reasonable accuracy. Any general purpose inference infrastructure, whether built on top of NIMI or RTCP, should have the flexibility to use both `mtrace` and the topology inference algorithms reported in [19, 4].

In addition to addressing the above deployment issues, we are also exploring new application areas for MINC. One, we are investigating extensions to our inference methodology to estimate link-level *delay* behavior. We have developed prototype estimators for the delay distribution and delay variance on internal links of a multicast tree based on end-to-end delay measurements. We will describe these results in future papers. Two, we believe our inference techniques will prove useful in reliable multicast applications. These applications need to aggregate receivers to achieve scalable loss recovery. MINC could be used to group receivers that are topologically close and share loss performance.

5 Related Work

A growing number of measurement infrastructure projects (e.g., AMP [1], Felix [6], IPMA [7],

NIMI [14], Surveyor [22], and Test Traffic [23]) aim to collect and analyze end-to-end performance data for a mesh of unicast paths between a set of participating hosts. We believe our multicast-based inference techniques would be a valuable addition to these measurement platforms. As mentioned in the previous section, we are working to incorporate MINC capabilities into NIMI.

A lot of recent experimental work has sought to understand internal network behavior from end-to-end performance measurements (e.g., see [5, 12, 17, 18]). In particular, `pathchar` [16] is under evaluation as a tool for inferring link-level statistics from end-to-end unicast measurements. Much work remains to be done in this area and with MINC we are contributing a novel multicast-based methodology.

Regarding multicast-based measurements, we have already described the `mtrace` tool [13]. In addition, the `tracer` tool [8] performs topology discovery through the use of `mtrace`. However, `mtrace` suffers from performance and applicability problems in the context of large-scale Internet measurements. First, as mentioned earlier in this paper, `mtrace` needs to run once for each receiver in order to cover a complete multicast tree. This behavior does not scale well to large numbers of receivers. In contrast, MINC covers the complete tree in a single pass. Second, `mtrace` relies on multicast routers to respond to explicit measurement queries. Although current routers support these queries, Internet Service Providers (ISPs) may choose to disable this feature since it gives anyone access to detailed delay and loss information about paths inside their networks. In contrast, MINC does not rely on cooperation from any network-internal elements.

6 Conclusions

We have presented experimental results that validate the MINC approach to inferring link-level loss rates from end-to-end multicast measurements. We compared loss rates in MBone tunnels as inferred using our technique and as measured by `mtrace`. Inferred values closely matched directly measured values – differences were usually well below 1%, never above 3%, while loss rates varied between 0 and 35%. In addition, our inference algorithm quickly converged to the true loss rates – inferred values stabilized well within 2-minute, 1200-probe measurement intervals.

We feel that MINC is an important new methodology for network measurement, particularly Internet measurement. It does not rely on network cooperation and it scales to very large networks. MINC is firmly grounded in statistical analysis that is backed up by packet-level simulations and now experiments under real network conditions. We are continuing to extend MINC along both analytical and experimental fronts.

Acknowledgements

Tian Bu wrote the `infer` program. Mark Handley suggested using RTCP receiver reports to carry MINC loss traces.

References

[1] AMP: Active Measurement Project. `http://amp.nlanr.net`

[2] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, "Multicast-Based Inference of Network-Internal Loss Characteristics," Comp. Sci. Tech. Rep. 98-17, University of Massachusetts at Amherst, February 1998. `ftp://gaia.cs.umass.edu/pub/CDHT98:MINC.ps.Z`

[3] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," *Proc. IEEE Infocom '99*, March 1999.

[4] R. Cáceres, N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley, "Loss-Based Inference of Multicast Network Topology," submitted for publication, February 1999.

[5] R. L. Carter, M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Proc. PERFORMANCE '96*, October 1996.

[6] Felix: Independent Monitoring for Network Survivability project. `ftp://ftp.bellcore.com/pub/mwg/felix/index.html`

[7] IPMA: Internet Performance Measurement and Analysis project. `http://www.merit.edu/ipma`

[8] B.N. Levine, S. Paul, J.J. Garcia-Luna-Aceves, "Organizing Multicast Receivers Deterministically According to Packet-Loss Correlation," *Proc. ACM Multimedia 98*, September 1998.

[9] `mbat`: MBone Analysis Tool. `ftp://gaia.cs.umass.edu/pub/mist`

[10] M. R. Macedonia, D. P. Brutzman, "MBone Provides Audio and Video across the Internet," *IEEE Computer*, April 1994.

[11] MINC: Multicast Inference of Network Characteristics project. `http://gaia.cs.umass.edu/minc`

[12] M. Mathis, J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, June 1996.

[13] `mtrace`: Print multicast path from a source to a receiver. `ftp://ftp.parc.xerox.com/pub/net-research/ipmulti`

[14] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, Vol. 36, No. 8, pp. 48-54, August 1998.

[15] `ns`: Network simulator. `http://www-mash.cs.berkeley.edu/ns`

[16] Pathchar: A Tool to Infer Characteristics of Internet Paths. `ftp://ftp.ee.lbl.gov/pathchar`

[17] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. ACM SIGCOMM '96*, August 1996.

[18] V. Paxson, "End-to-End Internet Packet Dynamics," *Proc. ACM SIGCOMM 1997*, September 1997.

[19] S. Ratnasamy, S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths Using End-to-End Measurements," *Proc. IEEE Infocom '99*, March 1999.

[20] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, January 1996.

[21] `sdr`: Session Directory tool. `http://www-mice.cs.ucl.ac.uk/multimedia/software/sdr`

[22] Surveyor project. `http://io.advanced.org/surveyor`

[23] Test Traffic project. `http://www.ripe.net/test-traffic`

Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation

R. Cáceres N.G. Duffield J. Horowitz D. Towsley T. Bu

Abstract—**We explore the use of end-to-end multicast traffic as measurement probes to infer network-internal characteristics. We have developed in an earlier paper [2] a Maximum Likelihood Estimator for packet loss rates on individual links based on losses observed by multicast receivers. This technique exploits the inherent correlation between such observations to infer the performance of paths between branch points in the multicast tree spanning the probe source and its receivers. We evaluate through analysis and simulation the accuracy of our estimator under a variety of network conditions. In particular, we report on the error between inferred loss rates and actual loss rates as we vary the network topology, propagation delay, packet drop policy, background traffic mix, and probe traffic type. In all but one case, estimated losses and probe losses agree to within 2 percent on average. We feel this accuracy is enough to reliably identify congested links in a wide-area internetwork.**

Keywords—**Internet performance, end-to-end measurements, Maximum Likelihood Estimator, tomography**

I. Introduction

A. Background and Motivation

Fundamental ingredients in the successful design, control and management of networks are mechanisms for accurately measuring their performance. Two approaches to evaluating network performance have been (i) collecting statistics at internal nodes and using network management packages to generate link-level performance reports; and (ii) characterizing network performance based on end-to-end behavior of point-to-point traffic such as that generated by TCP or UDP. A significant drawback of the first approach is that gaining access to a wide range of internal nodes in an administratively diverse network can be difficult. Introducing new measurement mechanisms into the nodes themselves is likewise difficult because it requires persuading large companies to alter their products. Also, the composition of many such small measurements to form a picture of end-to-end performance is not completely understood.

Regarding the second approach, there has been much recent experimental work to understand the phenomenology of end-to-end performance (e.g., see [1], [3], [15], [20], [22], [23]). A number of ongoing measurement infrastructure projects (Felix [6], IPMA [8], NIMI [14] and Surveyor [31]) aim to collect and analyze end-to-end measurements across a mesh of paths between a number of hosts. `pathchar` [11] is under evaluation as a tool for inferring link-level statistics from end-to-end point-to-point measurements. However, much work remains to be done in this area.

In a recent paper [2], we considered the problem of characterizing link-level loss behavior through end-to-end measurements. We presented a new approach based on the measurement and analysis of the end-to-end loss behavior of *multicast* probe traffic. The key to this approach is that multicast traffic introduces correlation in the end-to-end losses measured by receivers. This correlation can, in turn, be used to infer the loss behavior of the links within the multicast routing tree spanning the sender and receivers. Our principal analytical tool is a *Maximum Likelihood Estimator* (MLE) of the link loss rates. This estimate is derived under the assumption that link losses are described by independent Bernoulli losses. The data for this inference is a record of which of n probes were observed at each of the receivers. We have shown that these estimates are strongly consistent (converge almost surely to the true loss rates). Moreover, the asymptotic normality property of MLEs allows us to derive an expression for their rate of convergence to the true rates as n increases. The presence of spatial and temporal correlation between losses would violate the assumptions of the model. However, we showed in [2] that spatial correlations deform the Bernoulli based estimator continuously (i.e. small correlations give rise to only small inaccuracies). Moreover, the deformation is a second order effect in that it depends only on the change in loss correlations between different parts of the network. Temporal correlations do not alter the strong consistency of the estimator; they only slow the rate of convergence.

We envisage deploying inference engines as part of a measurement infrastructure comprised of hosts exchanging probes in a wide-area network (WAN). Each host will act as the source of probes down a multicast tree to the others. A strong advantage of using multicast rather than unicast traffic is efficiency. N multicast servers produce a network load that grows at worst linearly as a function of N. On the other hand, the exchange of unicast probes can lead to local loads which grow as N^2, depending on the topology.

B. Contribution

Whereas the experimental component of our previous work focused on comparing inferred and actual probe losses, the focus of this paper is on asking how close are the inferred losses to those of background traffic. We do this under a variety of network configurations. These are specified by varying the following: (i) network topology (ii) background traffic mix (iii) packet

This work was sponsored in part by the DARPA and Air Force Research Laboratory under agreement F30602-98-2-0238.

Ramon Cáceres is with AT&T Labs–Research, Rm. B125, 180 Park Avenue, Florham Park, NJ 07932, USA; E-mail: ramon@research.att.com

Nick Duffield is with AT&T Labs–Research, Rm. B139, 180 Park Avenue, Florham Park, NJ 07932, USA; E-mail: duffield@research.att.com

Joseph Horowitz is with the Dept. of Math. & Statistics, University of Massachusetts Amherst, MA 01003-4515, USA; E-mail: joeh@math.umass.edu

Don Towsley is with the Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA; E-mail: towsley@cs.umass.edu

Tian Bu is with the Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA; E-mail: tbu@cs.umass.edu

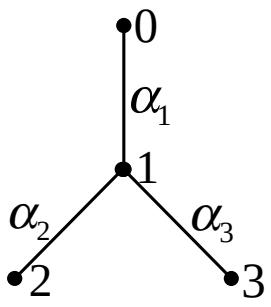


Fig. 1. A two-leaf logical multicast tree

drop policy (iv) probe traffic type, and (v) network propagation delay. In analyzing potential differences between inferred and actual losses we identify three potential causes.

The first is the statistical variability expected on the basis of the loss model. The general theory of MLE's furnished the asymptotic variance of the estimators as the number of probes grows. These tell us how many probes must be used in order to achieve measurements of a desired level of accuracy. It can be shown that the asymptotic variance of each estimated loss probability is, to first order, equal to the true loss probability and otherwise independent of the topology. The role of such theoretical values is to establish a baseline for variance of loss estimates of background traffic.

The second potential cause of differences is the non-conformance of probe losses to the Bernoulli model. In practice we find quite close agreement between inferred and actual probe losses. An examination of the underlying loss process shows that deviations from the Bernoulli model are quite small. The correlation between packet losses on different links is usually less than 0.1.

The main contribution to the difference comes from differences in the loss patterns exhibited by probe and background traffic. We have mainly used TCP background traffic in the simulations, reflecting the dominant use of TCP as a transport protocol on the Internet [32]. However, TCP flows are known to exhibit correlations. A well-known example of this is synchronization between TCP flows which can occur as a result of *slow start* after packet loss [10]. This mechanism can be expected to give rise to spatial and temporal correlations between losses. However, we believe that large and long-lasting spatial dependence is unlikely in a real network because of traffic heterogeneity. In our experiments we investigated the effects of two different discard methods: Drop from Tail and Random Early Detection (RED) [7]. One of the motivations for the introduction of RED has been to break dependence introduced through TCP.

The choice of probe process is one means by which we can aim to improve the accuracy of inference. A constraint on the interprobe time is that probe traffic should not itself contribute noticeably to congestion. Beyond the question of the mean, the choice of interarrival time distribution can affect the bias and variance of the MLE. Probes with exponentially distributed spacings will see time averages; this is the PASTA property (Poisson Arrivals See Time Averages; see e.g. [33]). This approach has been proposed for network measurements [24] and is under consideration in the IP Performance Metrics working group of the IETF [9]. We compare the effect of using constant rate probes and Poisson probes. In most cases the difference in accuracy is quite small. We find a far greater degradation in accuracy when network round trip times were reduced below the interprobe time.

The remaining sections of the paper are organized as follows. After a review of related work, in Section II we describe the loss model, in Section III the MLE and its properties. In Section IV we describe the algorithm used to compute the MLE from data. We discuss our framework for quantifying the errors in inference in Section V. The simulations themselves are reported in Section VI.

C. Related Work

In the opening paragraphs we listed a number of ongoing measurement infrastructure projects in progress ([6], [8], [14], [31]). We believe our multicast-based techniques would be a valuable addition to these measurement platforms.

Simultaneously with the present work, Ratnasamy and McCanne [26] have proposed using a multicast-based loss estimator to infer topology. The emphasis in their study is on grouping multicast receivers, rather than estimating the loss probabilities themselves. They use the same estimate as we do for loss on the shared path to two receivers, and this gives rise to an algorithm for inferring binary trees. Ad hoc extensions to trees with higher branching ratios are proposed.

There is a multicast-based measurement tool, `mtrace` [17], already in use in the Internet. `mtrace` reports the route from a multicast source to a receiver, along with other information about that path such as per-hop loss and delay statistics. Topology discovery through `mtrace` is performed as part of the `tracer` tool [13]. However, `mtrace` suffers from performance and applicability problems in the context of large-scale measurements. First, `mtrace` traces the path from the source to a single receiver by working back through the multicast tree starting at that receiver. In order to cover the complete multicast tree, `mtrace` needs to run once for each receiver, which does not scale well to large numbers of receivers. In contrast, the inference techniques described in this paper cover the complete tree in a single pass. Second, `mtrace` relies on multicast routers to respond to explicit measurement queries. Current routers support these queries. However, Internet service providers may choose to disable this feature since it gives anyone access to detailed delay and loss information about paths in their part of the network. (We have received reports that this is already occurring). In contrast, our inference techniques do not rely on cooperation from any network-internal elements.

There has been some ad hoc, statistically non-rigorous work on deriving link-level loss behavior from end-to-end multicast measurements. An estimator proposed in [34] attributes the absence of a packet at a set of receivers to loss on the common path from the source. However, this is biased, even as the number of probes n goes to infinity.

II. Description of the Loss Model

Let $\mathcal{T} = (V, L)$ denote the *logical* (as opposed to physical) multicast tree, consisting of the set of nodes V, including the source and receivers, and the set of links L, which are ordered pairs (j, k) of nodes, indicating a (directed) link from j to k. The set of *children* of node j is denoted by $d(j)$; these are the nodes with a link coming from j. For each node j, other than the root 0, there is a unique node $f(j)$, the *parent* of j, such that $j \in d(f(j))$. Each link can therefore be identified by its "child" endpoint. We define "ancestors" (grandparents and the like) in an obvious way, and likewise "descendants". The difference between a logical and a physical tree is that, whereas it is possible for a node to have only one child in the physical tree, in the logical tree each node except the root and leaves must have at least two children. A physical tree can be converted into a logical tree by deleting all nodes, other than the root, which have one child and adjusting the links accordingly.

The root $0 \in V$ represents the source of the probes and the set of *leaf* nodes $R \subset V$ (i.e., those with no children) represents the receivers.

A probe packet is sent down the tree starting at the root. If it reaches a node j a copy of the packet is produced and sent down the link toward each child of j. As a packet traverses a link k (recall that k denotes the endpoint), it is lost with probability $\overline{\alpha}_k = 1 - \alpha_k$ and arrives at k with probability α_k. We shall use the notation $\overline{\alpha} = 1 - \alpha$ for any quantity α (with or without subscripts) between 0 and 1. The losses on different links are assumed to be independent and to occur with the probabilities $\overline{\alpha}_k$ as described. In [2] we have discussed the potential limitations of this model, and how the model can be corrected if there are dependencies between the losses. The two-leaf logical multicast tree is shown in Figure 1.

We describe the passage of probes down the tree by a stochastic process $X = (X_k)_{k \in V}$ where each X_k equals 0 or 1: $X_k = 1$ signifies that a probe packet reaches node k, and 0 that it does not. The packets are generated at the source, so $X_0 = 1$. For all other $k \in V$, the value of X_k is determined as follows. If $X_k = 0$ then $X_j = 0$ for the children j of k (and hence for all descendants of k). If $X_k = 1$, then for j a child of k, $X_j = 1$ with probability α_j, and $X_j = 0$ with probability $\overline{\alpha}_j$, independently for all the children of k. We write $\alpha_0 = 1$ to simplify expressions concerning the α_k.

III. Maximum Likelihood Estimation of Loss

If a probe is sent down the tree from the source, the outcome is a record of whether or not a copy of the probe was received at each receiver. Expressed in terms of the process X, the outcome is a configuration $X_{(R)} = (X_k)_{k \in R}$ of zeroes and ones at the receivers (1 = received, 0 = lost). Notice that only the values of X at the receivers are observable; the values at the internal nodes are invisible. The state space of the observations $X_{(R)}$ is thus the set of all such configurations, $\Omega = \{0, 1\}^R$. For a given set of link probabilities $\alpha = (\alpha_k)_{k \in V}$, the distribution of $X_{(R)}$ on Ω will be denoted by P_α. The probability mass function for a single outcome $x \in \Omega$ is $p(x; \alpha) = \mathsf{P}_\alpha(X_{(R)} = x)$.

Let us dispatch n probes, and, for each $x \in \Omega$, let $n(x)$ denote the number of probes for which the outcome x is obtained. The probability of n independent observations $x^1, \ldots, x^n$ (with each $x^m = (x_k^m)_{k \in R}$) is then

$$p(x^1, \ldots, x^n; \alpha) = \prod_{m=1}^{n} p(x^m; \alpha) = \prod_{x \in \Omega} p(x; \alpha)^{n(x)} \qquad (1)$$

We estimate α using maximum likelihood, based on the data $(n(x))_{x \in \Omega}$, and we find that the usual regularity conditions that imply good large-sample behavior of the MLE are satisfied in the present situation. This is useful for the applications we have in mind because (a) we want to assess the accuracy of our estimates via confidence intervals, and (b) it is important to determine the smallest number n of probes needed to achieve the desired accuracy. We want to minimize n because, although sending out probes is inexpensive in itself, networks are subject to various fluctuations (e.g., [20]) which can perturb the model, and the measurement process itself ties up network resources.

We begin with a review of our main results on the existence and uniqueness of the MLE. Another question, not treated here, but which is important for applications, is the feasibility and organization of the computations. We work with the log-likelihood function

$$\mathcal{L}(\alpha) = \log p(x^1, \ldots, x^n; \alpha) = \sum_{x \in \Omega} n(x) \log p(x; \alpha). \qquad (2)$$

In the notation we suppress the dependence of $\mathcal{L}$ on n and $x^1, \ldots, x^n$. For each node k, let $\Omega(k)$ be the set of outcomes $x \in \Omega$ such that $x_j = 1$ for at least one receiver $j \in R$ which is a descendant of k, and let $\gamma_k = \Gamma_k(\alpha) := \mathsf{P}_\alpha[\Omega(k)]$. An estimate of γ_k is

$$\widehat{\gamma}_k = \sum_{x \in \Omega(k)} \widehat{p}(x), \qquad (3)$$

where $\widehat{p}(x) := n(x)/n$ is the observed proportion of trials with outcome x. We will show how to find α as a function of the γ. The MLE $\breve{\alpha}$ is precisely that α which maximizes $\mathcal{L}(\alpha)$:

$$\breve{\alpha} = \arg\max_{\alpha \in [0,1]^L} \mathcal{L}(\alpha) \qquad (4)$$

We shall see that, at least for large n, $\breve{\alpha} = \Gamma^{-1}(\widehat{\gamma})$, using the inverse of the function Γ that expresses the γ_k in terms of the α_k. Candidates for the MLE are solutions $\widehat{\alpha}$ of the *likelihood equation*:

$$\frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha) = 0, \quad k \in U. \qquad (5)$$

Set $\mathcal{A} = \{(\alpha_k)_{k \in U} : \alpha_k > 0\}$, and $\mathcal{G} = \{(\gamma_k)_{k \in U} : \gamma_k > 0 \, \forall k;\ \gamma_k < \sum_{j \in d(k)} \gamma_j \, \forall k \in U \setminus R\}$.

Theorem 1: When $\widehat{\gamma} \in \mathcal{G}$, the likelihood equation has the unique solution $\widehat{\alpha} := \Gamma^{-1}(\widehat{\gamma})$ that can be expressed as follows. Define $(\widehat{A}_k)_{k \in V}$ for the root node by $\widehat{A}_0 = 1$, for leaf nodes $k \in R$ by $\widehat{A}_k = \widehat{\gamma}_k$, and for all other nodes $k \in U \setminus R$ as the unique solution in $(0, 1]$ of

$$1 - \widehat{\gamma}_k / \widehat{A}_k = \prod_{j \in d(k)} (1 - \widehat{\gamma}_j / \widehat{A}_k). \qquad (6)$$

Then for $k \in U$, $\widehat{\alpha}_k = \widehat{A}_k / \widehat{A}_{f(k)}$.

The form (6) follows from the corresponding relations that express γ_k in terms of $A_k := \alpha_k \alpha_{f(k)} \ldots \alpha_0$.

We complete the picture by showing that the solution of the likelihood equation actually maximizes the likelihood function under some additional conditions. The set $\mathcal{A}$ contains all positive α_k, including the possibility $\alpha_k > 1$. Let us now restrict our attention to link probabilities $\alpha \in \mathcal{B} = (0,1)^{\#R} \subset \mathcal{A}$. Being a solution of the likelihood equation does not preclude $\widehat{\alpha}$ from being either a minimum or a saddlepoint for the likelihood function, with the maximum falling on the boundary of $\mathcal{B}$. For some simple topologies we are able to establish directly that $\mathcal{L}(\alpha)$ is (jointly) concave in the parameters at $\alpha = \widehat{\alpha}$, which is hence the MLE $\breve{\alpha}$. For more general topologies we use general results on maximum likelihood to show that $\widehat{\alpha} = \breve{\alpha}$ for all sufficiently large n.

Theorem 2:
(i) The model is identifiable in $\mathcal{B}$, i.e., $\alpha, \alpha' \in \mathcal{B}$ and $\mathsf{P}_\alpha = \mathsf{P}_{\alpha'}$ implies $\alpha = \alpha'$. Thus, distinct link probabilities α produce distinct statistical behavior of the $\widehat{\gamma}$ as $n \to \infty$.
(ii) As $n \to \infty$, $\breve{\alpha} \to \alpha$, with P_α- probability 1, i.e., the MLE is strongly consistent.
(iii) With probability 1, for sufficiently large n, $\breve{\alpha} = \widehat{\alpha}$, i.e., the solution of the likelihood equation maximizes the likelihood.
This is proven using large sample theory for MLE, such as in [30]. Finally we have a result on asymptotic normality of the MLE. The *Fisher Information Matrix* at α based on $X_{(R)}$ is the matrix $\mathcal{I}_{jk}(\alpha) := \mathsf{Cov}\left(\frac{\partial \mathcal{L}}{\partial \alpha_j}(\alpha), \frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha)\right)$.

Theorem 3: $\mathcal{I}(\alpha)$ is non-singular, and as $n \to \infty$, under P_α, $\sqrt{n}(\widehat{\alpha} - \alpha)$ converges in distribution to a multivariate normal random vector with mean vector 0 and covariance matrix $\mathcal{I}^{-1}(\alpha)$.

Example: MLE for the Two-Leaf Tree. Denote the 4 points of $\Omega = \{0,1\}^2$ by $\{00, 01, 10, 11\}$. Then

$$\widehat{\gamma}_1 = \widehat{p}(11) + \widehat{p}(10) + \widehat{p}(01), \tag{7}$$

$$\widehat{\gamma}_2 = \widehat{p}(11) + \widehat{p}(10), \quad \widehat{\gamma}_3 = \widehat{p}(11) + \widehat{p}(01), \tag{8}$$

and equations (6) for $\widehat{A}_k$ in terms of the $\widehat{\gamma}_k$ yield

$$\widehat{\alpha}_1 = \frac{\widehat{\gamma}_2 \widehat{\gamma}_3}{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1} \tag{9}$$

$$= \frac{(\widehat{p}(01) + \widehat{p}(11))(\widehat{p}(10) + \widehat{p}(11))}{\widehat{p}(11)} \tag{10}$$

$$\widehat{\alpha}_2 = \frac{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1}{\widehat{\gamma}_3} = \frac{\widehat{p}(11)}{\widehat{p}(01) + \widehat{p}(11)} \tag{11}$$

$$\widehat{\alpha}_3 = \frac{\widehat{\gamma}_2 + \widehat{\gamma}_3 - \widehat{\gamma}_1}{\widehat{\gamma}_2} = \frac{\widehat{p}(11)}{\widehat{p}(10) + \widehat{p}(11)} \tag{12}$$

Note that although it is possible that $\widehat{\alpha}_1 > 1$ for some finite n, this will not happen when n is sufficiently large, due to Theorem 2.

IV. Computation of the MLE on a General Tree

In this section we describe the algorithm for computing $\widehat{\alpha}$ on a general tree. An important feature of the calculation is that it can be performed recursively on trees. First we show how to calculate the $\widehat{\gamma}_k$. These can be calculated by reconstruction of a sample path of the full process $(X_k)_{k \in V}$ that is consistent with the measured data $X^1_{(R)}, \ldots, X^n_{(R)}$ from n probes. We define the n-element binary vector $(\widehat{X}_k)_{k \in V}$ recursively by

$$\widehat{X}_k = X_k, \quad k \in R \tag{13}$$

$$\widehat{X}_k(i) = \bigvee_{j \in d(k)} \widehat{X}_j(i), \quad k \in V \setminus R \tag{14}$$

so that

$$\widehat{\gamma}_k = n^{-1} \sum_{i=1}^{n} \widehat{X}_k(i). \tag{15}$$

For simplicity we assume now that $\widehat{\gamma} \in \Gamma((0,1)^{\#V})$. The calculation of $\widehat{\alpha}$ can be done by another recursion. We formulate both recursions in pseudocode in Figure 2. The procedure `find_x` calculates the $\widehat{X}_k$ and $\widehat{\gamma}_k$, assuming $\widehat{X}_k$ initializes to X_k for $k \in R$ and 0 otherwise. The procedure `infer` calculates the $\widehat{\alpha}_k$. The procedures could be combined. The full set of link probabilities is estimated by executing `main(1)`; recall 1 is the single descendant of the root node 0. Here, an empty product (which occurs when the first argument of `infer` is a leaf node) is understood to be zero. Here `solvefor` is a routine that finds the unique solution $\widehat{A}_k$ in $(0,1]$ to (6).

```
procedure main (k ) {
    find_x ( k ) ;
    infer ( k, 1 ) ;
}

procedure find_x ( k ) {
    foreach ( j ∈ d(k) ) {
        X̂_j = find_x ( j ) ;
        foreach ( i ∈ {1,...,n} ) {
            X̂_k[i] = X̂_k[i] ∨ X̂_j[i] ;
        }
    }
    γ̂_k = n^{-1} Σ_{i=1}^{n} X̂_k[i] ;
    return X̂_k ;
}

procedure infer ( k, A ) ;
    A_k = solvefor( A_k , (1 − γ̂_k / A_k) == Π_{j∈d(k)} (1 − γ̂_j / A_k) ) ;
    α̂_k = A_k / A ;
    foreach ( j ∈ d(k) ) {
        infer ( j , A_k ) ;
    }
}
```

Fig. 2. Pseudocode for Inference of Link Probabilities

The recursive nature of the algorithm has important consequences for its implementation in a network setting. The calculation of $\widehat{\gamma}_k$ and A_k depends on X only through the $(\widehat{X}_j)_{j \in d(k)}$. In a networked implementation this would enable the calculation to be localized in subtrees at a representative node. The computational effort at each node would be at worst proportional to the depth of the tree (for the node which is unlucky enough to be the representative for all distinct subtrees to which it belongs). The network load induced by the communication of data could be kept local, e.g., by scoped multicast amongst sibling representatives.

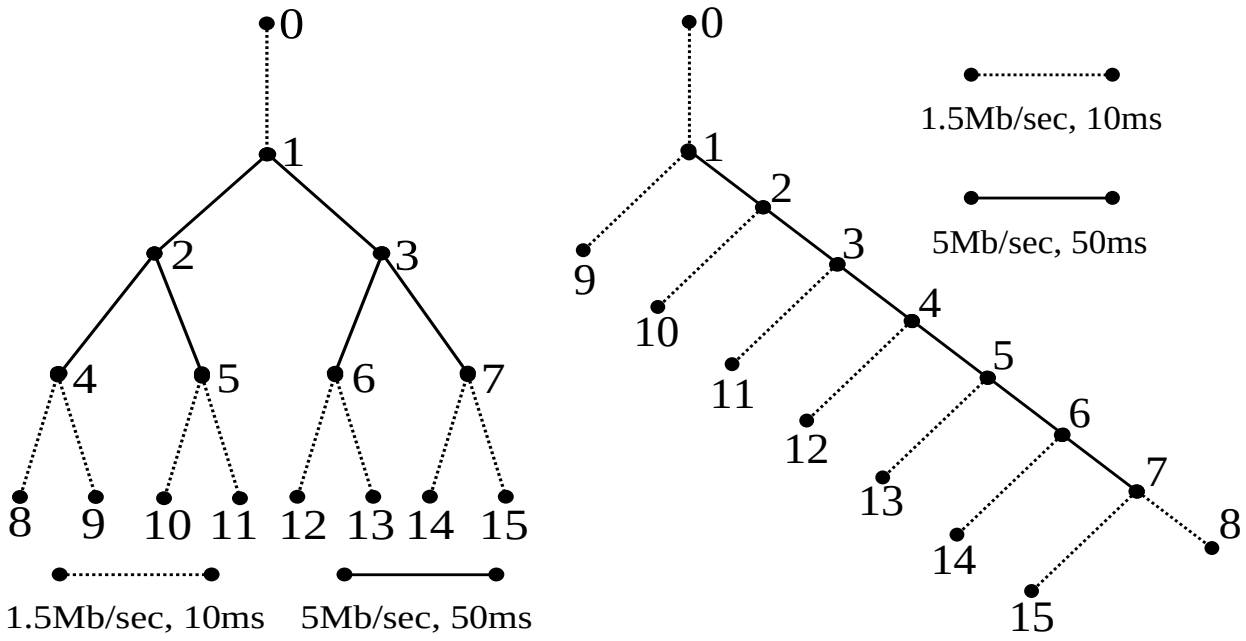


Fig. 3. SIMULATION TOPOLOGY: Links are of two types: "edge" links of 1.5Mb/s capacity and 10ms latency, and interior links of 5Mb/s capacity and 50ms latency. LEFT: "regular" topology with branching ratio 2. RIGHT: "irregular" topology.

V. FRAMEWORK FOR SIMULATION STUDY

We evaluated our loss inference algorithm using the `ns` simulator [19]. This enabled us to investigate the effectiveness of the estimator over a range of network topologies, link delays, packet drop policies, background traffic types, and probe traffic types. In particular we were able to determine the actual loss experienced by background traffic, and by probe traffic, and compare these values to those predicted by the inference algorithm on the basis of measurements at the leaf nodes. The experiments show that the agreement between inferred and probe loss is extremely good. This shows that the model of probe loss and the associated inference technique are quite effective in the small networks used in the simulation. This is encouraging since we expect flow synchronization effects (that would violate the model) to be more noticeable amongst a smaller numbers of flows. Agreement between inferred loss and background traffic loss is quite reasonable, although not as close as between inferred and probe loss. Some difference is expected due to the difference in temporal statistics of TCP flows and probes.

A. Comparing Loss Probabilities

We describe our approach to comparing two sets of loss probabilities p and q. For example p could be an inferred probability on a link, q the corresponding actual probability. For some **error margin** $\varepsilon > 0$ we define the **error factor**

$$F_\varepsilon(p,q) = \max\left\{\frac{p(\varepsilon)}{q(\varepsilon)}, \frac{q(\varepsilon)}{p(\varepsilon)}\right\} \tag{16}$$

where $p(\varepsilon) = \max\{\varepsilon, p\}$ and $q(\varepsilon) = \max\{\varepsilon, q\}$. Thus, we treat p and q as being not less than ε, and having done this, the error factor is the maximum ratio, upwards or downwards, by which they differ. Unless otherwise stated, we used the default value $\varepsilon = 10^{-3}$ in this paper. The choice of this metric is motivated by the expectation that it is desirable to estimate the relative magnitude of loss ratios on different links in order to distinguish those which suffer higher loss. In summarizing the relative accuracy of a set of loss measurements, we will calculate statistics of the error factor, such as mean and quantiles of $F_\varepsilon(p_i, q_i)$ where $p = (p_i)$ and $q = (q_i)$ are two sets of loss probabilities (inferred and actual, say). Here the index i runs over a set of links, a set of measurements on the same link made at different times or during different simulations, or some combination of these.

B. Summary Statistics of the Error Factor

In describing the mean and variability of the error factors, we shall use the following summary statistics. We shall estimate the center of the distribution of a set of error factors x_i by the two-sided quartile-weighted median

$$m(\{x\}) := (Q_{.25} + 2Q_{.5} + Q_{.75})/4 \tag{17}$$

where Q_p denotes the p^{th} quantile of the x_i. m is particularly suited to skewed distributions; see [29] for further detail. We characterize the high values of the error factors through the 90^{th} percentile. Both these summary statistics are robust, being independent of any assumption on the distribution of the error factors.

C. Experimental Variables

We explored the performance of the inference algorithm under variation of the following quantities.

C.1 Network Topology

We investigated three topologies. We used the two-leaf binary tree of Figure 1 to explore the variables listed below within a tightly controlled environment. We also explored two larger binary topologies: the regular 8 leaf binary tree of Figure 3(left), and the irregular tree of Figure 3(right). In both of the larger trees we arranged for some heterogeneity between the edges and the center in order to mimic the difference between the core

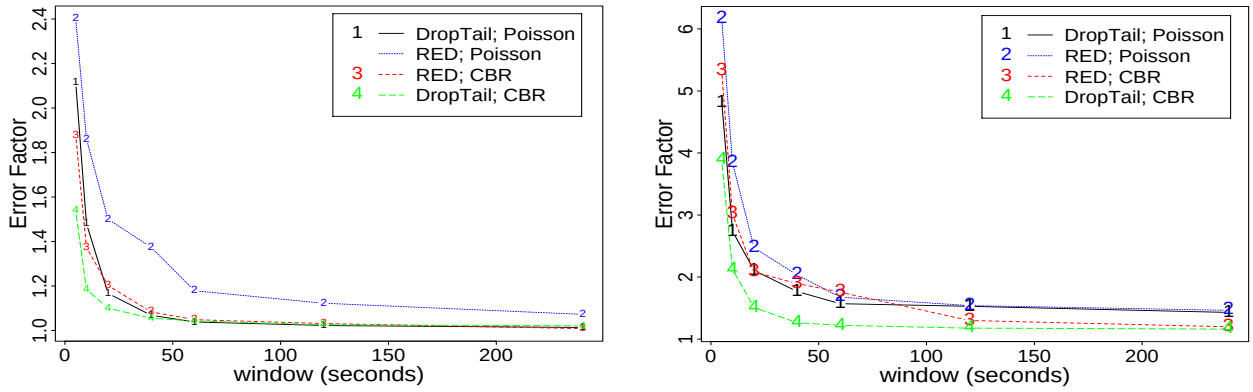


Fig. 4. ACCURACY OF INFERENCE VS. SAMPLE WINDOW: Mean error factor over all links and windows of regular topology in Figure 3(left) for RED or DropTail queueing; Poisson or CBR probes. LEFT: inferred loss vs. probe loss. RIGHT: inferred loss vs background loss. Probe bytes are 1.8% of of total; average utilization is 60%.

and edges of a large WAN, with the interior of the tree having higher capacity (5Mb/sec) and latency (50ms) than at the edge (1Mb/sec and 10ms).

C.2 Packet Discard Method

Each node had a buffer capacity of 20 packets, independent of packet size. We compare the effects of two methods of packet discard: Drop from Tail (DT), and discard based on Random Early Detection (RED) [7]. One of the benefits expected from the deployment of RED is increased utilization through the breaking of synchronization that can occur due to slow start of TCP after congestion, as identified in [10]. We used the `ns` default parameters of RED in the simulations.

C.3 Background Traffic

Each of the trees was equipped with a variety of flows of background traffic. Flows were of two types: infinite data sources that use the Transmission Control Protocol (TCP), and on-off sources using the Unreliable Datagram Protocol (UDP), the on and off periods having either a Pareto or an exponential distribution. In most of the simulations on the larger trees we used predominantly TCP, with a mixture of UDP. We chose this mix because TCP is the dominant transport protocol on the Internet [32].

C.4 Probe Characteristics

It is desirable that probe traffic only use a small part of the available link capacity. For the experiments in the large topologies we used 40-byte probes with a mean interprobe time of 16ms, i.e. a 20 kbit/sec stream. This is just over 1% of the capacity of the smallest link used; it would be a far smaller fraction of capacities commonly used in today's Internet backbones. We used two types of probes: constant rate probes and Poisson probes. The use of the latter has been proposed [24] for end-to-end measurements on the basis that Poisson Arrivals See Time Averages; see e.g. [33].

C.5 Relative Time Scales

We investigated the effects of network roundtrip time on estimator accuracy. This is potentially important because the roundtrip time determines the time it takes TCP to respond to packet losses. Thus the relative size of this time and the interprobe time determines the number of probe packets that sample congestion due to TCP traffic. In these experiments we reverted to a uniform link latency of between 1ms and 100ms.

VI. SIMULATION RESULTS

A. Qualitative Sample Path Behavior

We start by illustrating some properties of sample paths of the MLE. We shall make mostly qualitative observations initially; quantitative statistical measures of the accuracy of inference will be applied later.

In the regular topology of Figure 3(left) we conducted experiments of 240 seconds duration. Background traffic was generated by 30 infinite FTP sources using TCP, and another 30 on-off UDP sources, mostly with low rates and either exponential or Pareto distributed. There was one experiment for each of the four combinations of DropTail or RED packet discard and Poisson or CBR probes. The mean time between probes was 16ms, so about 15,000 probes were used in each experiment. For each of the experiments we calculated $\widehat{\alpha}$ on a moving window of a given width, using jumps of half the width. We display the mean error factor as a function of window size in Figure 4. On the left we show the error factor between inferred and actual probe loss; on the right between inferred and actual background loss. The main points to observe are that (i) error factors decrease as window size increases; (ii) the error factor between inferred and probe losses is small when compared with that between inferred and background losses; (iii) the error factors are reasonably insensitive to choice of packet discard method and probe type. To the extent that there are differences, mean error factors between inferred and background losses for CBR probes are slightly smaller than for Poisson probes, at least for larger window sizes (about 1.2 compared with about 1.5). Error factors for RED are marginally worse than for DropTail. We shall comment upon these differences later.

B. Dynamic Tracking of Loss

In Figure 5 we display the time series of background, probe and inferred loss on one link over the moving windows of a sim-

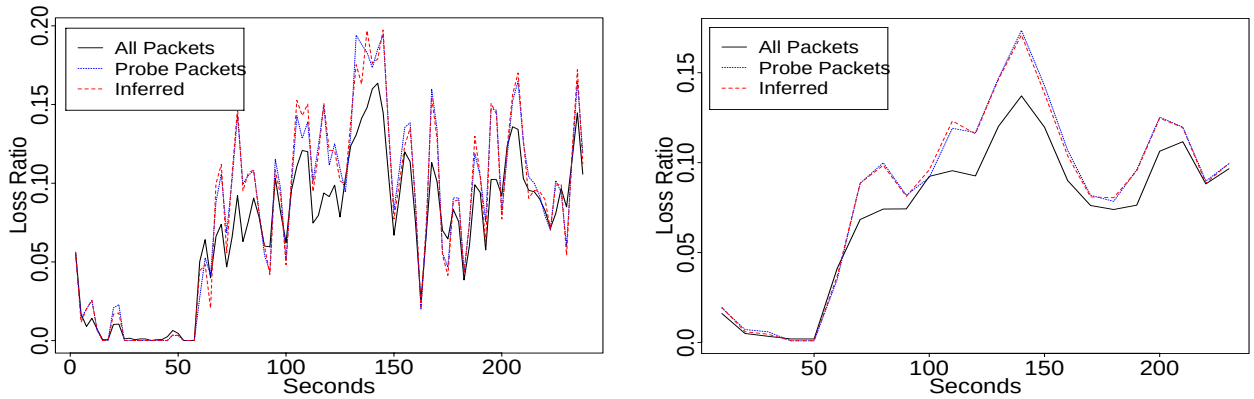


Fig. 5. DYNAMIC ACCURACY OF INFERENCE: Loss rates of background packets, probe packets and inferred on link 8 in regular topology in Figure 3(left) for RED queueing and Poisson probes. LEFT: 5 second window. RIGHT: 20 second window. Additional sources started at 60 seconds; note tracking by estimator of induced congestion. Probe bytes are 2% of total on 1.5Mb/s link with 60% utilization.

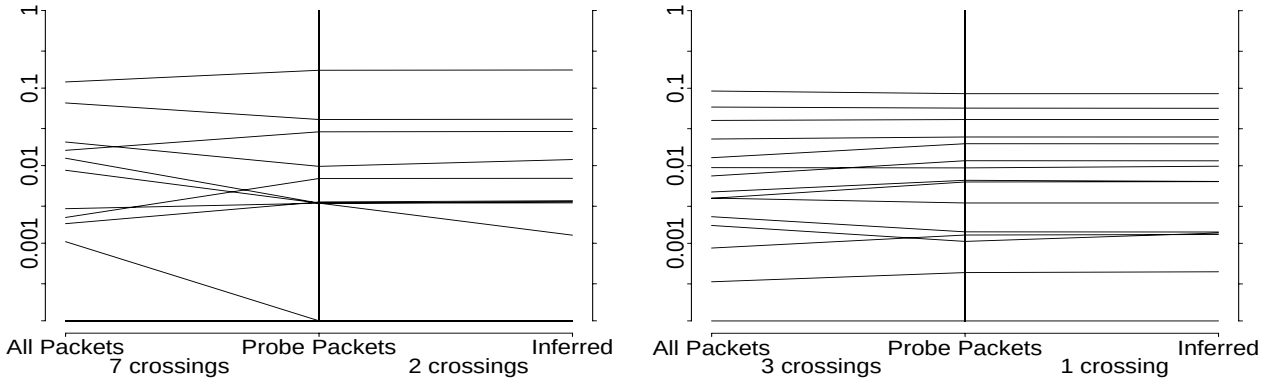


Fig. 6. ACCURACY AND ORDERING OF INFERENCE VS. SAMPLE WINDOW: Loss rates in regular topology of Figure 3(left) for RED queueing and Poisson probes. LEFT: 5 second window. RIGHT: 240 second window. Lines join probabilities of a given link. Fewer crossings indicate better preservation of order between actual and estimated probabilities. Flatter lines indicate better accuracy of estimates. Probe bytes are 3% of total on 1.5Mb/s link with 50% utilization.

ulation similar to that just described. However, we arrange for some additional sources to be turned on after 60 seconds have elapsed. We display how inferred losses track the real ones on a 5 second window (left) and a 20 second window (right). There is considerable variability between the inferred and actual loss at the 5 second window, not all of which is removed by increasing to a 20 second window. However, even at the 5 second window it appears that the estimator responds rapidly to the increase in actual loss that occurs after 60 seconds have elapsed.

From Figure 5 it is evident that the inferred loss tracks the probe loss more closely than the loss of background packets. Increasing the window size narrows some of the difference. We illustrate this for a single window in Figure 6. For a 5 second and a 240 second window, we display how the ordering of the links according to loss probability differs according to whether the loss used for ordering is that for background or probe or inferred loss. To do this we have placed each set of probabilities on an axis (background loss on left, probe loss in middle and inferred loss on right) and joined the values for given links. The flatter the lines, the greater the accuracy; the less they cross, the better the ordering is preserved. In this example, both accuracy and ordering are improved by using the larger window. It is clear in this example that despite error factors of about 2 between some of the inferred and background traffic losses, the inference is sufficiently accurate to distinguish the links with the highest loss for either probe or background packets.

C. Quantitative Statistical Measures of Accuracy

We now present some broad statistical measures of the accuracy of the inference in different network configurations in topologies with 15 links. We conducted 10 experiments of 240 seconds duration for each of the four combinations of DropTail or RED packet discard with CBR or Poisson probes. We then calculated the center m and 90^{th} percentile of the 150 error factors (10 experiments $\times$ 15 links).

The results are tabulated for the regular topology with mixed TCP and UDP sources in Table I; for the regular topology with TCP source only in Table II; and for the irregular topology with mixed sources in Table III. Taking these as a group, the accuracy of inference of probe loss is striking. Looking at the first pair of columns in each table we see that the error is no more than 2% of the true value on average (i.e. an error factor 1.02), the 90^{th} percentile of the error being 17% of the true value at worst (i.e. an error factor 1.17).

The error factors between actual probe loss and background traffic loss are somewhat larger; this difference is then the main contribution to errors in inferring the background traffic loss by the probe loss. The center m is less than 1.5, and the 90^{th} per-

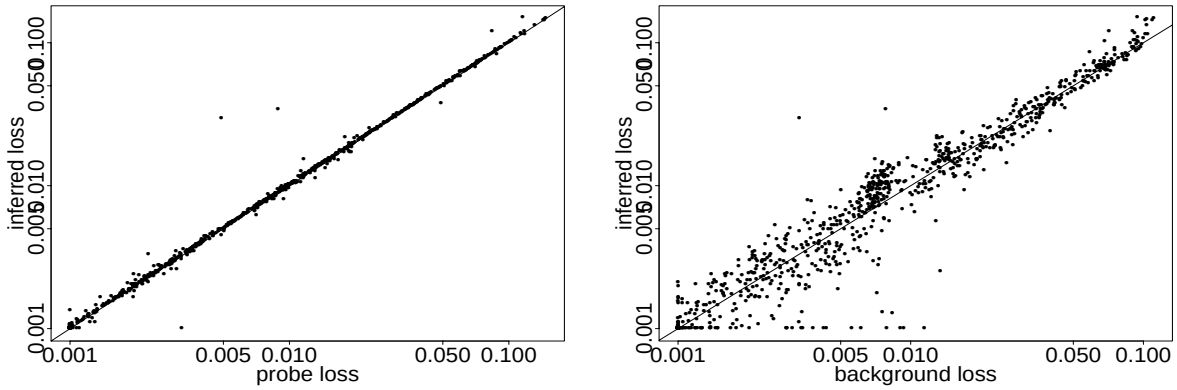


Fig. 7. ESTIMATOR ACCURACY: Scatter plots of 1110 pairs of loss probabilities gathered from all simulations: LEFT: inferred loss vs. probe loss; RIGHT: inferred loss vs. background loss. All probabilities truncated with error margin $\varepsilon = 10^{-3}$.

Discard Method	Probe Type	inf vs. probe		probe vs. b'grnd		inf vs. b'grnd	
		m	$Q_{.9}$	m	$Q_{.9}$	m	$Q_{.9}$
DT	PP	1.01	1.07	1.21	1.58	1.23	1.68
DT	CBR	1.00	1.03	1.11	1.43	1.11	1.43
RED	PP	1.01	1.04	1.14	1.54	1.15	1.56
RED	CBR	1.00	1.03	1.10	1.36	1.09	1.39

TABLE I

STATISTICS OF ERROR FACTOR VS. PACKET DISCARD AND PROBE METHOD. TCP and UDP background traffic. Regular Topology. Weighted Median and 90^{th} percentile of error factor over all links during 10 simulations of 240 seconds. Error margin was $\varepsilon = 10^{-3}$. In about 20% of cases, one or both probabilities compared were less than ε.

Discard Method	Probe Type	inf vs. probe		probe vs. b'grnd		inf vs. b'grnd	
		m	$Q_{.9}$	m	$Q_{.9}$	m	$Q_{.9}$
DT	PP	1.02	1.11	1.47	2.03	1.45	2.19
DT	CBR	1.01	1.06	1.31	1.82	1.33	1.81
RED	PP	1.01	1.06	1.42	1.92	1.43	1.91
RED	CBR	1.01	1.03	1.19	1.55	1.20	1.53

TABLE II

STATISTICS OF ERROR FACTOR VS. PACKET DISCARD AND PROBE METHOD. TCP background traffic only. Regular Topology. Weighted Median and 90^{th} percentile of error factor over all links during 10 simulations of 240 seconds. Error margin was $\varepsilon = 10^{-3}$. In about 15% of cases, one or both probabilities compared were less than ε.

Discard Method	Probe Type	inf vs. probe		probe vs. b'grnd		inf vs. b'grnd	
		m	$Q_{.9}$	m	$Q_{.9}$	m	$Q_{.9}$
DT	PP	1.02	1.17	1.34	1.83	1.39	2.24
DT	CBR	1.02	1.11	1.24	1.66	1.27	1.84
RED	PP	1.01	1.13	1.18	1.62	1.23	1.74
RED	CBR	1.01	1.08	1.13	1.54	1.17	1.61

TABLE III

STATISTICS OF ERROR FACTOR VS. PACKET DISCARD AND PROBE METHOD. TCP and UDP background traffic. Irregular Topology. Mean and 90^{th} percentile of error factor over all links during 10 simulations of 240 seconds. Error margin was $\varepsilon = 10^{-3}$. In no more than 8% of cases, one or both probabilities were less than ε.

Link Delay	inf vs. probe		probe vs. b'grnd		inf vs. b'grnd	
	m	$Q_{.9}$	m	$Q_{.9}$	m	$Q_{.9}$
100ms	1.00	1.04	1.07	1.45	1.07	1.44
30ms	1.00	1.02	1.17	1.54	1.17	1.53
10ms	1.09	1.71	1.28	1.88	1.19	1.49
1ms	1.49	6.83	1.30	1.61	1.71	5.07

TABLE IV

STATISTICS OF ERROR FACTOR VS. LINK DELAY. TCP and UDP background traffic. Regular Topology. DropTail with Poisson Probes. Weighted Median and 90^{th} percentile of error factor over all links during 10 simulations of 240s for each delay value. One or both probabilities compared were less than error margin $\varepsilon = 10^{-3}$ in up to 40% of cases.

centile is less than 2.2. Pure TCP background traffic has somewhat higher error factors than mixed TCP and UDP. The irregular topology has somewhat higher error factors than the regular topology. The average utilization in these simulations was about 60%. We also conducted simulations at up to 90% utilization on the two-leaf binary tree with approximately the same number of probes. In most cases the summary statistics were of the same order.

Comparing the different packet discard methods, we see that RED always gives somewhat lower values for m and the 90^{th} percentile than the corresponding DropTail. This fits with our expectation that the randomization induced by RED will break correlations induced by TCP flow control, and hence cause patterns of loss for background traffic to more closely resemble the Bernoulli loss model.

Comparing the different packet probe types, we see that CBR has m and 90^{th} percentile consistently slightly lower than for Poisson probes. The reason for this small difference is not clear at present. Poisson probes see time averages [33] and hence yield unbiased measurements. It is possible though that they exhibit higher variances for the reason that the potentially extreme (long or short) interarrival times lead to worse sampling of network congestion events.

We examined the influence of network propagation delay on error factors. For DropTail packet discard and Poisson queueing, we find (see Table IV) that error factors increase as propagation delay decreases. A possible explanation for this is the following. We observe an increase in utilization as the propagation delay is decreased, the utilization being close to 100% on some links when propagation delay is 1ms. Since recovery after TCP losses will be correspondingly quick, any spare capacity will be rapidly exploited, and congestion may be long lived, leading to temporal correlations between probe losses. Whereas this would not alter the asymptotic accuracy of the MLE, it would slow the rate of convergence as the number of probes is increased, leading to high estimator variance. This hypothesis is supported by

Table IV: at 1ms feedback delay, most of the error is between the inferred and probe loss. 1ms is far shorter than the minimum link propagation delays on the Internet, so we do not expect this phenomenon to occur in practice. We stress, however, that it remains to obtain a full understanding of the effect on accuracy of the interactions between interprobe time, propagation delay and variables such as packet discard method and probe type.

We summarize all our experiments hitherto in Figure 7, where we show a scatter plot of pairs of (inferred loss, probe loss) on the left, and pairs of (inferred loss, background loss) on the right. Thus each point corresponds to a single link on a single simulation run. Also included here are points for experiments conducted with the combinations of traffic types, discard method, probe distribution and topology described above, but with a more variable flow duration. The flow durations were obtained by choosing random beginning and end times for each flow in a given simulation, rather than having the flows present for the whole simulation. In these examples, inferred loss is a better predictor of background loss when the latter is at least 1%: for this subset of data points the mean error factor is 1.20 compared with 1.28 for the complete set.

VII. Conclusions

In this paper we have analyzed the efficacy of multicast-based inference in estimating loss probabilities in the interior of a network from end-to-end measurements. The principal tool was a Maximum Likelihood Estimator of the link loss probabilities. Probes are multicast from a source; the data for the MLE is a record of which probes were received at each leaf of the multicast tree. Although the method assumes that losses are independent, we have shown in some cases that it is relatively insensitive to the presence of spatial loss correlations; temporal correlations increase its variance, so that a longer measurement period is required; see [2].

We evaluated the method by conducting `ns` simulations that used topologies and traffic flows with quite a rich structure, with several hops per flow and flows per link. We compare inferred and actual loss probabilities on the links of the logical multicast tree. The experiments showed that the loss probabilities for probe packets were inferred extremely closely by the MLE.

The probe traffic was typically only 1% to 2% of the traffic on each link. We investigated how closely loss rates for background traffic were inferred. We examined the effect of changing traffic mix, topology, packet discard method and probe type. We found small differences between these, compared with the inherent variability of the estimates. Varying the network feedback delay also affected the accuracy of inference. For very short propagation delays we believe that the aggressive behavior of TCP slow start is a factor in decreasing accuracy. We intend to investigate this phenomenon more fully.

Over a range of experiments our summary statistics show that the relative error of the inferred and actual losses had a distribution whose center was no greater than about 1.5 and whose 90^{th} percentile was no worse than a factor of about 2.2. If one is limited to using inferred probe loss to estimate background traffic loss, this would mean that only in 1% of the worst cases would a *single* inference fail to distinguish between two background loss rate that separated by a factor of 5. We believe that this is sufficiently accurate to identify the most congested links.

References

[1] J-C. Bolot and A. Vega Garcia "The case for FEC-based error control for packet audio in the Internet" ACM Multimedia Systems, to appear.

[2] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, "Multicast-based inference of network-internal characteristics", Comp. Sci. Tech. Rep. 98-17, University of Massachusetts at Amherst, February 1998. `ftp://gaia.cs.umass.edu/pub/CDHT98:MINC.ps.Z`

[3] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *PERFORMANCE '96*, October 1996.

[4] A. Dembo and O. Zeitouni, "Large deviations techniques and applications", Jones and Bartlett, Boston, 1993.

[5] B. Effron and D.V. Hinkley, "Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information", Biometrika, 65, 457–487, 1978.

[6] Felix: Independent Monitoring for Network Survivability. For more information see `ftp://ftp.bellcore.com/pub/mwg/felix/index.html`

[7] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, 1(4), August 1993.

[8] IPMA: Internet Performance Measurement and Analysis. For more information see `http://www.merit.edu/ipma`

[9] IP Performance Metrics Working Group. For more information see `http://www.ietf.org/html.charters/ippm-charter.html`

[10] V. Jacobson, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM '88, August 1988, pp. 314–329.

[11] V. Jacobson, Pathchar - A Tool to Infer Characteristics of Internet paths. For more information see `ftp://ftp.ee.lbl.gov/pathchar`

[12] E.L. Lehmann. "Theory of point estimation". Wiley-Interscience, 1983.

[13] B.N. Levine, S. Paul, J.J. Garcia-Luna-Aceves, "Organizing multicast receivers deterministically according to packet-loss correlation", Preprint, University of California, Santa Cruz.

[14] J. Mahdavi, V. Paxson, A. Adams, M. Mathis, "Creating a Scalable Architecture for Internet Measurement," *to appear in Proc. INET '98*.

[15] M. Mathis and J. Mahdavi, "Diagnosing Internet Congestion with a Transport Layer Performance Tool," *Proc. INET '96*, Montreal, June 1996.

[16] S.P. Meyn and R.L. Tweedie, "Markov chains and stochastic stability", Springer, New York, 1993.

[17] `mtrace` – Print multicast path from a source to a receiver. For more information see `ftp://ftp.parc.xerox.com/pub/net-research/ipmulti`

[18] `nam` – Network Animator. For more information see `http://www-mash.cs.berkeley.edu/ns/nam.html`

[19] `ns` – Network Simulator. For more information see `http://www-mash.cs.berkeley.edu/ns/ns.html`

[20] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. SIGCOMM '96*, Stanford, Aug. 1996.

[21] V. Paxson, "Towards a Framework for Defining Internet Performance Metrics," *Proc. INET '96*, Montreal, 1996.

[22] V. Paxson, "End-to-End Internet Packet Dynamics," *Proc. SIGCOMM 1997*, Cannes, France, 139–152, September 1997.

[23] V. Paxson, "Automated Packet Trace Analysis of TCP Implementations," *Proc. SIGCOMM 1997*, Cannes, France, 167–179, September 1997.

[24] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. Dissertation, University of California, Berkeley, April 1997.

[25] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[26] S. Ratnasamy & S. McCanne, "Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements", Proceedings IEEE Infocom'99, New York, (1999).

[27] K. Ross & C. Wright, "Discrete Mathematics", Prentice Hall, Englewood Cliffs, NJ, 1985.

[28] W. Rudin, "Functional Analysis", McGraw-Hill, New York, 1973.

[29] L. Sachs, "Applied Statistics", Springer, New York, 1982.

[30] M.J. Schervish, "Theory of Statistics", Springer, New York, 1995.

[31] Surveyor. For more information see `http://io.advanced.org/surveyor/`

[32] K. Thompson, G.J. Miller and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, 11(6), November/December 1997.

[33] R.R. Wolff "Poisson Arrivals See Time Averages", Operations Research, 30: 223–231, 1982

[34] M. Yajnik, J. Kurose, D. Towsley, "Packet Loss Correlation in the MBone Multicast Network," *Proc. IEEE Global Internet*, Nov. 1996